DM34x0 MODBUS INTERFACE DEFINITION

CONTENTS

1 INTRODUCTION

It is possible to communicate via MODBUS with the DM3410 and DM3420 indicators when the RS485 pod is fitted into an option slot.

Modbus is a Master-Slave based communications protocol that means that all messages may only be initiated by the Master device. In general the Master will communicate with one Slave device at a time, although it is possible under certain circumstances for the Master to broadcast to the entire network.

The DM34x0 units are Slave devices and therefore require to be put on a network that has a Master in order to operate. This guide contains sufficient information in order to program and configure the Master Modbus device so that parameters from the DM34x0 units may be accessed.

2 DM34X0 CONFIGURATION

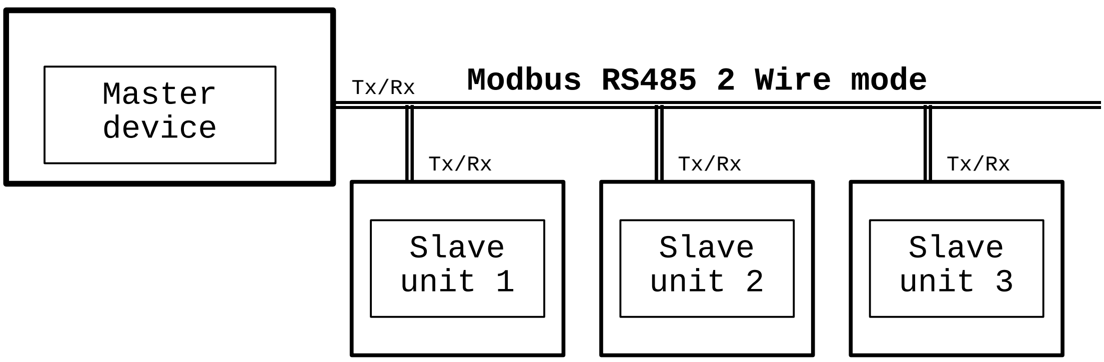
When the comms option pod is fitted to the unit from power-up, there will be a communications menu available. There are three items to configure.

Slave Baud rate (bAud)- 19.2kB or 9.6kB available
Mode (Line) - 2 wire or 4 wire available
Device Address (Addr) - Network unique address 1-255
(Note that maximum device no. for MODBUS is 247)

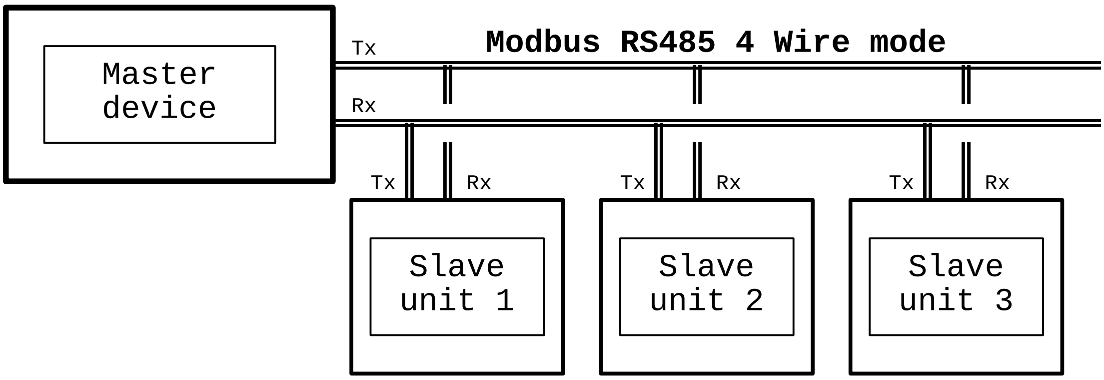
The comms port settings of 1 stop bit, 1 start bit, 8 data bits and no parity is fixed and therefore may not be changed.

The baud rate should be set up for the network. Although it is theoretically possible to set different baud rates for different devices it is recommended that one baud rate is chosen throughout.

The RS485 mode determines the way that the network is connected together. The 2-wire arrangement, shown below, has both transmit and receive signals sharing the same wires. Although this makes most efficient use of the connections and makes wiring simpler, correct operation depends upon critical timing within the Master device. A reply from a Slave device will occur about 5ms after the Master has issued a message, therefore, the Master output buffer needs to be disabled in time to prevent a data clash and a corrupted reply from the Slave.

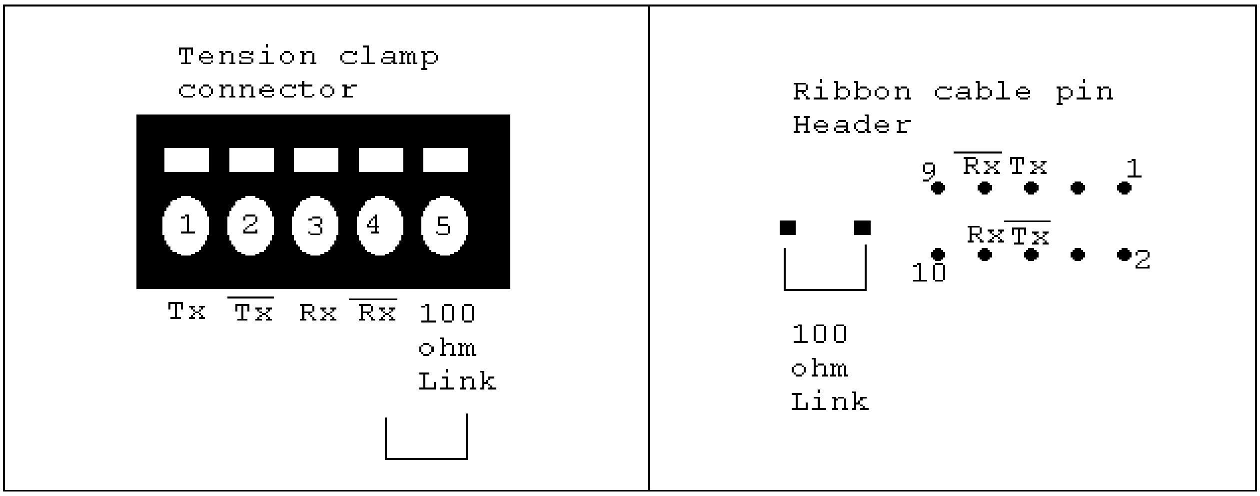


If 4-wire mode is chosen, the network is wired as shown below and does not have the same critical requirements for Master device message timing as the Master has dedicated transmit lines.



Device addresses
Each Slave unit requires a unique address to be programmed. If two or more units have the same address on the network, both or all will respond when this address is accessed by the Master device and a data corruption will result. Possible addresses range from 1-255; however, Modbus defines a maximum address number of 247. The electrical characteristics of RS485 limit the number of devices on a network to 32; however, buffering the network increases this number.

Electrical Interface
Two styles of connection are available:-
Tension clamp connector and ribbon cable interface. The pin-out assignments are shown below.



RS485 requires that the extreme ends of a network be terminated with resistances. A terminating resistor is incorporated within the comms pod and may be connected by fitting a link as indicated.

3 THE MODBUS PROTOCOL

The basic Modbus RTU protocol format for both Master and Slaves is as follows:

Slave Address	Function Code	Data	Error Check

The Slave address is as described above.
The Function code defines the type of comms operation.
There are four function codes supported.

Function code	**Description**
3	Read Register(s) value
16	Write Register(s) value
65	Request Unit id
66	Perform Slave action

The Data field is Function code dependent and is detailed in the following sections.

The CRC (Cyclic Redundancy Check) is a 16 bit field transmitted alongside the message and is used to confirm its accuracy. The method of calculation of the CRC is contained in the Appendix to this document.

3.1 FUNCTION 3 READ REGISTER

The general format for the read register request/reply sequence is as follows (each cell represents an eight bit byte):

Request issued from Master.

Slave Address	**3**	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	CRC Low	CRC High

The Start Addr High and Low make up a 16 bit register address word. Note that the index of the register allocation table is based from 1 whilst the value that is entered in the comms message is zero based. For example, the address of the Process Variable value stated as ‘1’ in the Register allocation table is entered as 0x0000 in the Modbus comms field.

The Number of registers requested is limited to 8 due to comms buffer restrictions. Therefore the “No. Regs High” field is always set to zero.

A single register represents a 16-bit data field and therefore in order to access a floating-point number, two consecutive registers need to be requested. The format of the number returned is IEEE 754 floating point format. It follows that the maximum number of floating point values that can be requested in a message of this type is limited to 4.

Format of the reply from Slave.

Slave Address	**3**	No. of Bytes	Reg_n High value	Reg_n Low value		CRC Low	CRC High

The “No. Of Bytes” value represents the number of bytes of actual data returned. As each register is a 16-bit number, the number of bytes value is the number of registers requested times two.

Function 3 Example
The Process Variable and Input Type are to be requested from a DM3410 addressed as device 6. The following message is transmitted by the Master

Addr	Func^{n}	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	CRC Low	CRC High
6	3	0	0	0	3		

Note that Start Address is zero based and that three registers are requested, two for the floating-point process variable and one for the input type register.

Although all Slave devices on the network 'hear' the message, only device 6 responds.

Addr	Func^{n}	No. of Bytes	Reg$_{n}$ High value	Reg$_{n}$ Low value	Reg$_{n+1}$ High value	Reg$_{n+1}$ Low value	Reg$_{n+2}$ High value	Reg$_{n+2}$ Low value	CRC Low	CRC High
6	3	6	0x41	0xC9	0x00	0x00	0x00	0x01		

This indicates that:

Process Variable value	= 0x41C90000	= 18.25
Input type	= 0x0001	= Thermocouple

3.2 FUNCTION 16 WRITE REGISTER

The general format for the write register request/reply sequence is as follows (each cell represents an eight bit byte):

Request issued from Master.

Slave Addr	**16**	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	No. Bytes	Reg_n High value	Reg_n Low value		CRC Low	CRC High

The Start Addr High and Low make up a 16 bit register address word. Note that the index of the register allocation table is based from 1 whilst the value that is entered in the comms message is zero based.

The Number of registers to be written to is limited to 8 due to comms buffer restrictions. Therefore the “No. Regs High” field is always set to zero.

Format of the reply from Slave.

Slave Address	**16**	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	CRC Low	CRC High

Function 16 Example

The Alarm Setpoint A in Slot 1 on device 111 (0x6F) is to be set to 100.0. The register address for this is 59; this corresponds to a zero based address of 58 (0x3A) for the comms command. Note that for consistency all the byte values are expressed in hexadecimal format.

Addr	Func^n	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	No. of Bytes	Reg_n High value	Reg_n Low value	Reg_{n+1} High value	Reg_{n+1} Low value	CRC Low	CRC High
0x6F	0x10	0x00	0x3A	0x00	0x02	0x04	0x00	0x01	0x00	0x01		

After setting the alarm setpoint to 100, device 111 sends the following message in way of acknowledgement.

Addr	Func^n	Start Addr High	Start Addr Low	No. Regs High	No. Regs Low	CRC Low	CRC High
0x6F	0x10	0x00	0x3A	0	2		

3.3 FUNCTION 65 REQUEST SLAVE ID

This function is used by the Master to determine the identification of the Slave device.
Request issued from Master.

Slave Address	**65**	Code High	Code Low	CRC Low	CRC High

The Code High and Low is not used and should be set to zero.

Format of the reply from Slave.

Slave Address	**65**	No. of Bytes (6)	Product Type	Product Variant	S/W Issue 0	S/W Issue 1	S/W Issue 2	S/W Issue 3	CRC Low	CRC High

The number of bytes reflects the number of data items in the reply and is set to 6.

The product type codes are as follows:

INDICATOR = 0x01
TRANSMITTER = 0x02

The Product Variant codes depend upon the product type. The codes for the indicator series is as follows:

DM3410 = 10_{10}
DM3420 = 20_{10}

The four-byte software code represents the issue date of the instrument software.

3.4 FUNCTION 66 SLAVE ACTION

This function is used by the Master to cause the Slave to perform some action.

Message issued from Master.

Slave Address	**66**	Code High	Code Low	CRC Low	CRC High

The type of action is determined by the Code number.
Code High is always set to zero.

Code value	**Slave Action**
0	Save Configuration data to E^2PROM
1	Save Calibration and Configuration data to E^2PROM
2	Save User defined Area to E^2PROM
3	Save RTX data to RTX E^2PROM

Format of the reply from Slave.

Slave Address	**65**	No. of Bytes (6)	Product Type	Product Variant	S/W Issue 0	S/W Issue 1	S/W Issue 2	S/W Issue 3	CRC Low	CRC High

The number of bytes reflects the number of data items in the reply and is set to 6.

The product type codes are as follows:

INDICATOR = 0x01
TRANSMITTER = 0x02

The Product Variant codes depend upon the product type.
The codes for the indicator series is as follows:

DM3410 = 10_{10}
DM3420 = 20_{10}

The four-byte software code represents the issue date of the instrument software.

4 MODBUS REGISTER ALLOCATION

The following tables denote the register allocation for DM3410 and DM3420 instruments.

The register numbering convention begins from 1 rather than zero, however the register addresses referenced in Section 3 require zero based addresses.

The format of the data is defined as follows: -

Float	These variables are 4 byte IEEE754 format floating point numbers that require two consecutive registers. Each register carries the most significant and least significant integer respectively.
Integer List	This is an integer register value which contains an index which relates to a sequential list of options contained within the ‘Units’ heading. The first item in the list being zero.
Bitmap	This is an integer value for which specific bits have defined functions.

4.1 DM3410 REGISTER ALLOCATION

4.1.1 GENERAL PARAMETERS

REG	Description	Format	Units	Notes
1	Process Variable	Float	°C, °F	
2	Process Variable (ls)	-	-	
3	Input type	Integer List	rtd, tc	
4	Linearisation index	Integer List	Euro,DIN,JISC,X	1,9
4	Linearisation index	Integer List	K.J,T,R,S,E,F,N,X	2,9
5	Temperature Units	Integer List	°C, °F	
6	System Status	Bitmap	See Note 7	
7	Not Allocated	-	-	
8	Not Allocated	-	-	
9	Not Allocated	-	-	
10	Not Allocated	-	-	
11	Input type	Integer List	rtd, tc	
12	Linearisation index	Integer List	Euro,DIN,JISC,X	1,9
12	Linearisation index	Integer List	K.J,T,R,S,E,F,N,X	2,9
13	Temperature Units	Integer List	°C, °F	
14	Burn Out	Integer List	High, Low	
15	Filter Factor	Integer List	none,2s,10s Adaptive	
16	Display Resolution	Integer List	xxxx,xx.x,x.xx,x.xxx	
17	Menu type	Integer List	Short, Full	
18	Enable Clear Latches	Integer List	False,true	
19	Enable Setpoint Edit	Integer List	False,true	
20	User Offset	Float	Engineering Units	
21				
22	Passcode	Integer	0 to 9999	
23	Not Allocated	-	-	
24	Not Allocated	-	-	
25	Not Allocated	-	-	
26	Not Allocated	-	-	
27	Not Allocated	-	-	
28	Not Allocated	-	-	
29	Not Allocated	-	-	
30	Not Allocated	-	-	

4.1.2 SLOT 1 PARAMETERS

REG	Description	Format	Units	Notes
31	Device No. Slot1	Integer	0-255	3,6
32	TAG Char1 & Char 2	2*8bit Ascii	-	3
33	TAG Char3 & Char 4	2*8bit Ascii	-	3
34	TAG Char5 & Char 6	2*8bit Ascii	-	3
35	TAG Char7 & Char 8	2*8bit Ascii	-	3
36	TAG Char9 & Char 10	2*8bit Ascii	-	3
37	Option slot contents	Integer List	empty, relay,rtx,comms	
38	Relay A status	Bitmap	See Note 8	4
39	Setpoint A	Float	°C, °F	4
40				4
41	Hysteresis A	Float	°C, °F	4
42				4
43	Relay B status	Bitmap	See Note 8	4
44	Setpoint B	Float	°C, °F	4
45				4
46	Hysteresis B	Float	°C, °F	4
47				4
38	Retransmission status	Integer List	4-20,0-20,0-10mA	5
39	Retransmission low	Float	°C, °F	5
40				5
41	Retransmission High	Float	°C, °F	5
42				5
43	Rtx cal. checksum	Integer	0-255	5,10
44	Cal Rtx Low	Float	mA	5,10
45				5,10
46	Cal Rtx High	Float	mA	5,10
47				5,10
48	Not Allocated	-	-	
49	Not Allocated	-	-	
50	Not Allocated	-	-	

4.1.3 SLOT 2 PARAMETERS

REG	Description	Format	Units	Notes
51	Device No. Slot1	Integer	0-255	3,6
52	TAG Char1 & Char 2	2*8bit Ascii	-	3
53	TAG Char3 & Char 4	2*8bit Ascii	-	3
54	TAG Char5 & Char 6	2*8bit Ascii	-	3
55	TAG Char7 & Char 8	2*8bit Ascii	-	3
56	TAG Char9 & Char 10	2*8bit Ascii	-	3
57	Option slot contents	Integer List	empty, relay,rtx,comms	
58	Relay A status	Bitmap	See Note 8	4
59	Setpoint A	Float	°C, °F	4
60				4
61	Hysteresis A	Float	°C, °F	4
62				4
63	Relay B status	Bitmap	See Note 8	4
64	Setpoint B	Float	°C, °F	4
65				4
66	Hysteresis B	Float	°C, °F	4
67				4
58	Retransmission status	Integer List	4-20,0-20,0-10mA	5
59	Retransmission low	Float	°C, °F	5
50				5
61	Retransmission High	Float	°C, °F	5
62				5
63	Rtx cal. checksum	Integer	0-255	5,10
64	Cal Rtx Low	Float	mA	5,10
65				5,10
66	Cal Rtx High	Float	mA	5,10
67				5,10
68	Not Allocated	-	-	
69	Not Allocated	-	-	
70	Not Allocated	-	-	

4.1.4 DIAGNOSTIC DATA

REG	Description	Format	Units	Notes
71	100 ohm cal value	Float	100r rdg for calibration	
72				
73	50mV cal value	Float	50mV rdg for calibration	
74				
75	CJ cal value	Float	CJ calibration offset	
76				
77	Calibration checksum	Integer	0-255	
78	Not allocated			
79	Not allocated			
80	Not allocated			
81	Isolate word	Bitmap	See 11 and Appendix ?	
82	Raw Rtd resistance	Float	Ω	
83				
84	Raw Lead r	Float	Ω	
85				
82	Raw mV input	Float	mV	
83				
84	Raw cj temp	Float	°C	
85				
86	Cal’d RTD	Float	Ω	
87				
88	Cal’d Lead r	Float	Ω	
89				
90	Led Data Buff0	Bitmap		
91	Led Data Buff1	Bitmap		
92	Led Data Buff2	Bitmap		
93	Demanded Current	Real	mA	
94				
95	PWM Mark time	Real	µS	
96				

4.2 DM3420 MODBUS REGISTER ALLOCATION

4.2.1 GENERAL PARAMETERS

REG	Description	Format	Units	Notes
1	Process Variable	Float	Engineering Units	
2				
3	Input type	Integer List	current, voltage	
4	Linearisation index	Integer List	Linear, sqrt, Custom	
5	Input Range	Integer List	4-20, 0-20, 0-10	13
5	Input Range	Integer List	1-5, 0-1, 0-10	14
6	System Status	Bitmap	See Note 7	
7	Not Allocated	-	-	
8	Not Allocated	-	-	
9	Not Allocated	-	-	
10	Not Allocated	-	-	
11	Input type	Integer List	current, voltage	
12	Linearisation index	Integer List	Linear, sqrt, Custom	
13	Input Range	Integer List	4-20, 0-20, 0-10	13
13	Input Range	Integer List	1-5, 0-1, 0-10	14
14	Burn Out	Integer List	High, Low	
15	Filter Factor	Integer List	none,2s,10s Adaptive	
16	Display Resolution	Integer List	xxxx, xx.x, x.xx, x.xxx	
17	Menu type	Integer List	Short, Full	
18	Enable Clear Latches	Integer List	False, true	
19	Enable Setpoint Edit	Integer List	False, true	
20	User Offset	Float	Engineering Units	
21				
22	Passcode	Integer	0 to 9999	
23	Engineering Low	Float	Engineering Units	
24				
25	Engineering High	Float	Engineering Units	
26				
27	Not Allocated	-	-	
28	Not Allocated	-	-	
29	Not Allocated	-	-	
30	Not Allocated	-	-	

4.2.2 SLOT 1 PARAMETERS

As the DM3410.

4.2.3 SLOT 2 PARAMETERS

As the DM3410.

4.2.4 DIAGNOSTIC DATA

REG	Description	Format	Units	Notes
71	20mA cal value	Float	20mA rdg for calibration	
72				
73	1V cal value	Float	1V rdg for calibration	
74				
75	10 cal value	Float	10V rdg for calibration	
76				
77	Calibration checksum	Integer	0-255	
78	Not allocated			
79	Not allocated			
80	Not allocated			
81	Isolate word	Bitmap	See 11 and Appendix ?	
82	Raw Analogue input	Float	mA or Volts	
83				
84	Not Allocated			
85				
86	Cal'd Analogue input	Float	mA or Volts	
87				
88	Not Allocated			
89				
90	Led Data Buff0	Bitmap		
91	Led Data Buff1	Bitmap		
92	Led Data Buff2	Bitmap		
93	Demanded Current	Real	mA	
94				
95	PWM Mark time	Real	µS	
96				

Notes

1. Applicable when input type is set to RTD.
2. Applicable when input type is set to Thermocouple.
3. Applicable only for a comms option pod
4. Applicable only for a relay option pod
5. Applicable only for a current rtx option pod.
6. Valid device numbers reside between 1 and 255. However maximum device number specified by the MODBUS specification is 247.
7. System Status Bitmap
   - Bit0 - Input Sensor Fault
   - Bit1 - Sensor Under-range
   - Bit2 - Sensor Over-range
   - Bit3 - Calibration data fault
   - Bit6 - ADC Processor not ready

8. Alarm Status Bitmap
   - Bit0,1 - Alarm type(00-Off;01-High;10-Low;11-On)
   - Bit2 - Latch Alarm (1- Alarm is latched)
   - Bit3 - Invert Energisation (1-Reversed)
   - Bit7 - Alarm Status (1- Alarm triggered)

   Note that Bits 4,5 and 6 are undefined.

9. Type X linearisation requires that the User Defined Area is correctly programmed (See Appendix?)
10. The current calibration data represents the milliamps generated for a mark time of 4096uS and 8192uS respectively. The overall cycle time is 16.384mS. The calibration data byte is calculated such that a modulo 256 sum of the bytes in the rtx cal data PLUS the calibration checksum results in 0xAA.
11. The Isolate Bitmap is always cleared on power up and has the following bit allocation.
   - Bit0 - Isolate Raw Input
   - Bit1 - Isolate RTX
   - Bit2 - Isolate PWM
   - Bit3 - Not allocated
   - Bit4 - Isolate Alarms
   - Bit5 - Isolate Display
   - Bit6 - Not Allocated
   - Bit7 - Isolate PV

12. TBA
13. Applicable when input type is selected to current.
14. Applicable when input type is selected to voltage.

5 CRC CALCULATION

Message checking is provided using a Cyclic Redundancy Check value that is calculated by the transmitting device and appended to the message. The receiving device recalculates the CRC and compares it with the appended CRC generated by the transmitter. If there has been any corruption of the message, the two CRC values will be very unlikely to match.

The CRC is started by first pre-loading a 16 bit register to all 1's. Then a process begins of applying successive 8 bit bytes of the message to the current contents of the register. Only the 8 bits of data in each character are used for generating the CRC. Start and Stop bits and the parity bit if one is used do not apply to the CRC.

During the generation of the CRC, each 8 bit character is exclusively ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is exclusively ORed with a pre-set fixed value of 0xA001. If the LSB is 0, no exclusive OR takes place.

This process is repeated until 8 shifts have been performed. After the last (eighth) shift, the next 8-bit byte is exclusively ORed with the registers current value and the process repeats for eight more shifts as described above. The final contents of the register, after all of the bytes of the message have been applied is the CRC value.

6 USER DEFINED AREA

The User Defined area (UDA) is a memory block which may be used for a range of functions. The entire memory area is downloaded from E^2ROM memory on power-up and may by modified and saved to E^2ROM using Modbus messages.

UDA Addr	**Modbus Reg'r**	**UDA used as scratch-pad**	**UDA used as linear interpl'n**	**UDA used as 4^{th} order polynomial**	**UDA used as 15^{th} order polynomial**
0x00	0x4000	UDAD=0x00	UDAD=0x10	UDAD=0x14	UDAD=0x1F
0x01					
0x02	0x4001		X0	X0	X0
0x03					
0x04	0x4002				
0x05			Y0	A0	A0
0x06	0x4003				
0x07					
0x08	0x4004		X1	B0	B0
0x09					
0x0A	0x4005				
0x0B			Y1	C0	C0
0x0C	0x4006				
0x0D					
0x0E	0x4007		X2	D0	D0
0x0F					
0x10	0x4008	General Purpose Scratchpad			
0x11			Y2	E0	E0
0x12	0x4009				
0x13					
0x14	0x400A		X3	X1	F0
0x15					
0x16	0x400B				
0x17			Y3	A1	G0
0x18	0x400C				
0x19					
0x1A	0x400D		X4	B1	H0
0x1B					
0x1C	0x400E				
0x1D			Y4	C1	I0
0x1E	0x400F				
0x1F					
0x20	0x4010		X5	D1	J0
0x21					
0x22	0x4011				
0x23			Y5	E1	K0
0x24	0x4012				
0x25					
0x26	0x4013		X6	X2	L0
0x27					
0x28	0x4014				
0x29			Y6	A2	M0
0x2A	0x4015				
0x2B					
0x2C	0x4016		X7	B2	N0
0x2D					
0x2E	0x4017				
0x2F			Y7	C2	O0
0x30	0x4018				
0x31					
0x32	0x4019		X8	D2	P0
0x33					
0x34	0x401A				
0x35			Y8	E2	X1
0x36	0x401B				
0x37					
0x38	0x401C		X9	X3	
0x39					
0x3A	0x401D				
0x3B			Y9		
0x3C	0x401E				
0x3D					
0x3E	0x401F		Not Used	mV/°C	mV/°C
0x3F					
0x40	0x4020				
0x41					
0x42	0x4021		Text area to tag linearisation characteristic		
0x43					
0x44	0x4022				
0x45					

The function of the UDA is defined by the first byte within the memory block, the UDA Definition (UDAD) byte. If the UDAD is set to zero, the UDA may be used to store whatever data the user may require. A typical application would be to store a few lines of text to record calibration or configuration information.

If the top nibble of UDAD is set to 0x1 then the function of the UDA is to perform linearisation upon the electrical input. The type of linearisation is defined by the least significant nibble in UDAD. If the LS nibble is zero the unit performs a linear interpolation upon the X/Y coordinate points sequentially arranged within the UDA area. If the electrical input is less than X0, an input underange is flagged. If the electrical input is greater than X9, a input overrange is flagged. The co-ordinate points may be freely allocated as long as the X co-ordinates are arranged in increasing value order. A six byte tag is included in this area in order to identify the characteristic.

If UDAD ls nibble is set to a nonzero value between 1 and 15, the unit performs a polynomial expansion based upon the data contained in the UDA. The order of polynomial is taken as the value of the nibble. In order to acheive greater flexibility and accuracy, the characteristic may be segmented and an individual polynomial series allocated specifically to each segment. Each set of polynomial coefficients has a low limit at the start and a high limit at the end. The instrument will accept as many polynomial segments that may be fitted within the UDA.

ISOLATION OF SYSTEM FUNCTIONS