



Energy Division

Integra 1630 Communications Guide

Contents

1	Integra 1630 Modbus™ implementation	3
1.1	Modbus™ Overview	3
1.2	Input Registers	3
1.3	Modbus™ Holding Registers and Integra set up	5
2	RS485 General Information	8
2.1	Half Duplex	8
2.2	Connecting the Instruments	9
2.3	A and B terminals	9
2.4	Troubleshooting	10
3	MODBUS General Information	11
3.1	MODBUS Message Format	11
3.2	Serial Transmission Modes	12
3.3	MODBUS Message Timing (RTU Mode)	13
3.4	How Characters are Transmitted Serially	13
3.5	Error Checking Methods	14
3.5.1	Parity Checking	14
3.5.2	CRC Checking	14
3.6	Function Codes	15
3.7	IEEE floating point format	15
3.8	MODBUS Commands supported	17
3.8.1	Read Input Registers	17
3.8.2	Read Holding Registers	18
3.8.3	Write Holding Registers	19
3.9	Exception Response	19
3.9.1	Table of Exception Codes	20
3.10	Diagnostics	20
4	Modbus™ TCP (Ethernet)	21
4.1	Communication Parameters	21
4.2	IP Address Assignment	21
4.2.1	Connections for configuring the IP address	21
4.2.2	Configuring a PC for Ethernet Integra	22
4.3	Single Register Response	26
5	BACnet IP interface	27
5.1	Introduction	27
5.2	Communication Parameters	27
5.3	IP Address Assignment	27
5.4	Initialisation and register identification.	27
5.5	Supported Queries	28
5.6	Protocol Implementation Conformance Statement	28
5.7	Changing the BACnet Node ID	30
5.8	Procedure For Changing The Node-Name	31
6	RS485 Implementation of Johnson Controls Metasys	32
6.1	Application details	32
6.1.1	Metasys release requirements	32
6.1.2	Support for Metasys Integration	32
6.1.3	Support for Crompton Integra operation	32
6.1.4	Design considerations	32
6.2	METASYS N2 Integra Point Mapping table	33
7	Integra Profibus Interface	35
7.1	GSD file	35
7.2	Floating Point Format	35
7.3	Single Parameter access	35
7.4	Functionality of the PLC Function Block	35
7.4.1	Reading	35
7.4.2	Writing	36
7.5	Common Problems	36
7.6	Modules available	36

1 Integra 1630 Modbus™ implementation

1.1 Modbus™ Overview

This section provides basic information for interfacing the Integra to a Modbus™ network. If background information or more details of the Integra implementation is required please refer to section 2 and 3 of this document.

Integra offers the option of an RS485 communication facility for direct connection to SCADA or other communications systems using the Modbus™ RTU slave protocol. The Modbus™ protocol establishes the format for the master's query by placing into it the device address, a function code defining the requested action, any data to be sent, and an error checking field. The slave's response message is also constructed using Modbus™ protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurs in receipt of the message, Integra will make no response. If the Integra is unable to perform the requested action, it will construct an error message and send it as the response.

The electrical interface is 2-wire RS485, via 3 screw terminals. Connection should be made using twisted pair screened cable (Typically 22 gauge Belden 8761 or equivalent). All "A" and "B" connections are daisy chained together. The screens should also be connected to the "Gnd" terminal. To avoid the possibility of loop currents, an Earth connection should be made at only one point on the network.

Line topology may or may not require terminating loads depending on the type and length of cable used. Loop (ring) topology does not require any termination load.

The impedance of the termination load should match the impedance of the cable and be at both ends of the line. The cable should be terminated at each end with a 120 ohm (0.25 Watt min.) resistor.

A total maximum length of 3900 feet (1200 metres) is allowed for the RS485 network. A maximum of 32 electrical nodes can be connected, including the controller.

The address of each Integra can be set to any value between 1 and 247. Broadcast mode (address 0) is not supported.

The maximum latency time of an Integra is 150ms i.e. this is the amount of time that can pass before the first response character is output. The supervisory programme must allow this period of time to elapse before assuming that the Integra is not going to respond.

The format for each byte in RTU mode is:

Coding System: 8-bit per byte

Data Format: 4 bytes (2 registers) per parameter.
Floating point format (to IEEE 754)

Most significant register first (Default). The default may be changed if required -
See Holding Register "Register Order" parameter.

Error Check Field: 2 byte Cyclical Redundancy Check (CRC)

Framing: 1 start bit

8 data bits, least significant bit sent first

1 bit for even/odd parity (or no parity)

1 stop bit if parity is used; 1 or 2 bits if no parity

Data Coding

All data values in the INTEGRA 1630 are transferred as 32 bit IEEE 754 floating point numbers, (input and output) therefore each INTEGRA value is transferred using two Modbus registers. All register read requests and data write requests must specify an even number of registers. Attempts to read/write an odd number of registers prompt the INTEGRA to return a Modbus exception message.

The INTEGRA 1630 can transfer a maximum of 40 values in a single transaction, therefore the maximum number of registers requestable is 80. Exceeding this limit prompts the INTEGRA 1630 to generate an exception response.

Data Transmission speed is selectable between 4800, 9600, 19200 and 38400 baud.

1.2 Input Registers

Input registers are used to indicate the present values of the measured and calculated electrical quantities. Each parameter is held in two consecutive 16 bit registers. The following table details the 3X register address, and the values of the address bytes within the message. A tick (√) in the column indicates that the parameter is valid for the particular wiring system. Any parameter with a cross (X) will return the value Zero. Each parameter is held in the 3X registers. Modbus™ Function Code 04 is used to access all parameters.

Address (Register)	Parameter Number	Parameter	Modbus™ Start Address Hex		3 Ø	3 Ø	1 Ø
			Hi Byte	Lo Byte	4 W	3 W	2 W
30251	126	THD Current Mean	00	FA	√	√	√
30253	127	Hours Run	00	FC	√	√	√
30255	128	Power Factor (+Ind/-Cap)	00	FE	√	√	√
30259	130	Current 1 Demand	01	02	√	√	√
30261	131	Current 2 Demand	01	04	√	√	X
30263	132	Current 3 Demand	01	06	√	√	X
30265	133	Current 1 Max. Demand	01	08	√	√	√
30267	134	Current 2 Max. Demand	01	0A	√	√	X
30269	135	Current 3 Max. Demand	01	0C	√	√	X

1.3 Modbus™ Holding Registers and Integra set up

Holding registers are used to store and display instrument configuration settings. All holding registers not listed in the table below should be considered as reserved for manufacturer use and no attempt should be made to modify their values.

The holding register parameters may be viewed or changed using the Modbus™ protocol. Each parameter is held in two consecutive 4X registers. Modbus™ Function Code 03 is used to read the parameter and Function Code 16 is used to write. Write to only one parameter per message.

Address (Register)	Parameter Number	Parameter	Modbus™ Start Address Hex		Valid range	Mode
			High Byte	Low Byte		
40001	1	Demand Time	00	00	Write: 0 but see * below.	r/w
40003	2	Demand Period	00	02	8,15,20,30, 60 minutes.	r/w
40007	4	System Voltage	00	06	1V - 400kV	r/wp
40009	5	System Current	00	08	1A - 9999 A	r/wp
40011	6	System Type	00	0A	See below for values.	r/wp
40013	7	Relay Pulse Width	00	0C	3,5 or 10 (x20ms)	r/w
40015	8	Energy Reset	00	0E	Write: 0 to reset.	wo
40019	10	RS485 set-up code	00	12	See table below.	r/w
40021	11	Node Address	00	14	1 - 247	r/w
40023	12	Relay Pulse Divisor	00	16	1,10,100,1000	r/w
40025	13	Password	00	18	0000 - 9999	r/w
40037	19	System Power	00	24		ro
40041	21	Register Order	00	28	Write: 2141 only.	r/w
40043	22	High Serial Number	00	2A	0 - 16,777,215	ro
40045	23	Low Serial Number	00	2C	0 - 16,777,215	ro
40057	29	Max Pulse Relay Setups	00	38	2	ro
40059	30	Selected Pulse Relay	00	3A	1 - 2	r/w
40061	31	Selected Energy Param.	00	3C	0, 37 - 41	r/wp
40099	50	Hours Run Reset	00	62	Write: 0 to reset.	wo
40101	51	Hours Run VA Level	00	64	0.0 – 0.5	r/w
40299	150	Secondary Volts	01	2A	Min Vin - Max Vin	r/wp
40307	154	Max Energy Count	01	32	6,7 or 8 digits	r/wp

r/w = read/write r/wp = read and write with password clearance ro = read only wo = write only

Some registers marked wo above may in fact be read, but the value returned is not valuable.

It is perfectly feasible to change Integra set-up using a general purpose Modbus™ master, but often easier to use the Integra display or Integra configurator software, especially for gaining password protected access. The Integra configurator software has facilities to store configurations to disk for later retrieval and rapid set up of similarly configured products.

Password Settings marked r/wp require the instrument password to have been entered into the Password register before changes will be accepted. Once the instrument configuration has been modified, the password should be written to the password register again to protect the configuration from unauthorised or accidental change. Power cycling also restores protection. Reading the Password register returns 1 if the instrument is unprotected and 0 if it is protected from changes.

* **Demand Time** is used to reset the demand period. A value of zero must be written to this register to accomplish this. Writing any other value will cause an error to be returned. Reading this register after instrument restart or resetting demand period gives the number of minutes of demand data up to a maximum of the demand period setting. For example, with 15 minute demand period, from reset the value will increment from zero every minute until it reaches 15. It will remain at this value until a subsequent reset occurs.

Demand Period represents demand time in minutes. The value written must be one of 8,15, 20, 30 or 60. Writing any other value will cause an error to be returned.

System Voltage in a PT/VT connected system represents the PT/VT primary voltage. In a direct connected (i.e. no PT.VT) system this parameter should be set the same as secondary volts.

System Current is the CT primary current.

System Type is set to '1' for single phase 2 wire, '2' for 3 Phase 3 Wire or '3' for 3 Phase 4 Wire.

Relay Pulse Width is the width of the relay pulse in multiples of 20 ms. However, only values of 3 (60 ms), 5 (100 ms) or 10 (200 ms) are supported. Writing any other value will cause an error to be returned.

Reset Energy is used to reset the Energy readings. A value of zero must be written to this register to accomplish this. Writing any other value will cause an error to be returned.

RS485 Set-Up Code

Baud Rate	Parity	Stop Bits	Decimal Value
38400	NONE	2	30
38400	NONE	1	14
38400	ODD	1	13
38400	EVEN	1	12
19200	NONE	2	26
19200	NONE	1	10
19200	ODD	1	9
19200	EVEN	1	8
9600	NONE	2	22
9600	NONE	1	6
9600	ODD	1	5
9600	EVEN	1	4
4800	NONE	2	18
4800	NONE	1	2
4800	ODD	1	1
4800	EVEN	1	0

Exercise caution when attempting to change mode via direct Modbus™ writes. Use of a display or the Integra configurator software is recommended.

Node Address is the Modbus™ or JC N2 slave address for the instrument. Any value between 1 and 247 can be set.

Relay Pulse Divisor supports only values of 1,10,100 or 1000. For example, with a relay pulse divisor of 1 and an energy parameter of 'Import kWh', each relay pulse will represent 1 kWh import. The value of divisor that can be written is automatically limited by the maximum pulse rate of 2 pulses per second at 144% of rated power.

System Power is the maximum system power based on the values of system type, system volts and system current.

Register Order controls the order in which the Integra receives or sends floating-point numbers: - normal or reversed register order. In normal mode, the two registers that make up a floating point number are sent most significant register first. In reversed register mode, the two registers that make up a floating point number are sent least significant register first. To set the mode, write the value '2141.0' into this register - the instrument will detect the order used to send this value and set that order for all Modbus™ transactions involving floating point numbers.

Max Pulse Relay Setups is the maximum number of pulse relays that can be setup in the instrument. The actual number of pulse relays fitted will be this number or less, see instrument rating label.

Selected Pulse Relay is the number of the pulse relay, 1 or 2, selected for editing the energy parameter number.

Selected Energy Param. is the energy parameter number, see below, for the selected pulse relay. Setting the parameter to zero will disable the relay.

Parameter Number	Energy Parameter
0	Relay Disabled
37	Import kWh sum
38	Export kWh sum
39	Import kVArh sum
40	Export kVArh sum
41	kVAh sum

Hours Run VA Level is the proportion of rated VA that is necessary for the hours run parameter to start accumulating time. Hours run VA level can range from 0.0 to 0.5 in steps of 0.002. The default value is 0.1, i.e. 10% of rated VA, a value of 0.0 will cause the hours run to operate continuously.

Secondary Volts indicates the voltage on the VT secondary when the voltage on the Primary is equal to the value of System Volts . The value of this register can be set to between the minimum and maximum instrument input voltage.

Maximum Energy Count controls the number of digits the energy (kWh and kvarh) counters can use before they roll over (i.e. resets to zero). The values of 6, 7 or 8 can be written to this register to indicate the number of digits to use. Other values will be rejected. Note, the display can only show 7 digits. If the energy exceeds 7 digits then the display will indicate overload with dashes, i.e. “- - - - -”.

2 RS485 General Information

Some of the information in this section relates to other (i.e. non 1630) Integra product families, and is included to assist where a mixed network is implemented.

RS485 or EIA (Electronic Industries Association) RS485 is a balanced line, half-duplex transmission system allowing transmission distances of up to 1.2 km. The following table summarises the RS-485 Standard:

PARAMETER	
Mode of Operation	Differential
Number of Drivers and Receivers	32 Drivers, 32 Receivers
Maximum Cable Length	1200 m
Maximum Data Rate	10 M baud
Maximum Common Mode Voltage	12 V to -7 V
Minimum Driver Output Levels (Loaded)	+/- 1.5 V
Minimum Driver Output Levels (Unloaded)	+/- 6 V
Drive Load	Minimum 60 ohms
Driver Output Short Circuit Current Limit	150 mA to Gnd, 250 mA to 12 V 250 mA to -7 V
Minimum Receiver Input Resistance	12 kohms
Receiver Sensitivity	+/- 200 mV

Further information relating to RS485 may be obtained from either the EIA or the various RS485 device manufacturers, for example Texas Instruments or Maxim Semiconductors. This list is not exhaustive.

2.1 Half Duplex

Half duplex is a system in which one or more transmitters (talkers) can communicate with one or more receivers (listeners) with only one transmitter being active at any one time. For example, a "conversation" is started by asking a question, the person who has asked the question will then listen until he gets an answer or until he decides that the individual who was asked the question is not going to reply.

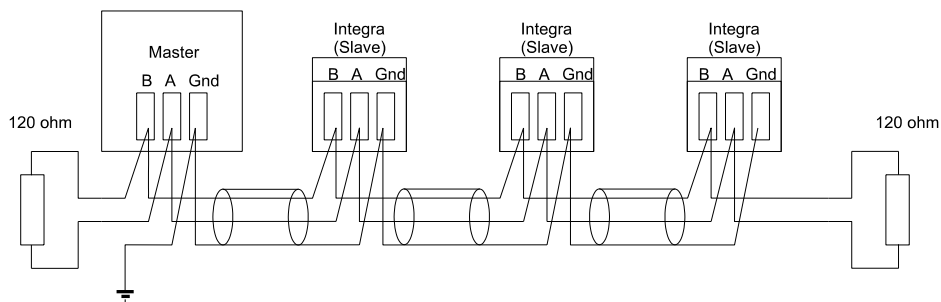
In a 485 network the "master" will start the "conversation" with a "query" addressed to a specific "slave", the "master" will then listen for the "slave's" response. If the "slave" does not respond within a pre-defined period, (set by control software in the "master"), the "master" will abandon the "conversation".

2.2 Connecting the Instruments

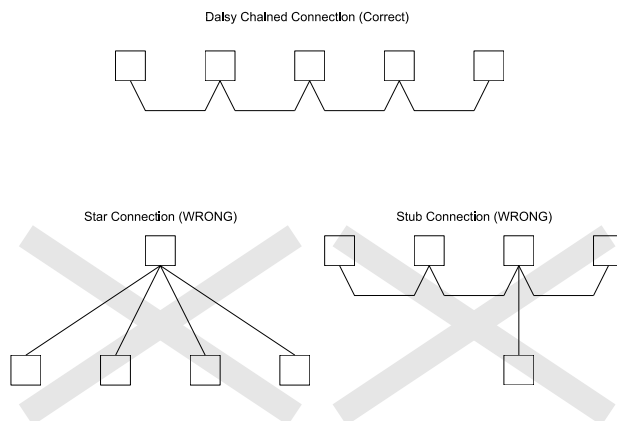
If connecting an RS485 network to a PC use caution if contemplating the use of an RS232 to 485 converter together with a USB to RS485 adapter. Consider either an RS232 to RS485 converter, connected directly to a suitable RS232 jack on the PC, or use a USB to RS485 converter or, for desktop PCs a suitable plug in RS485 card. (*Many 232:485 converters draw power from the RS232 socket. If using a USB to RS232 adapter, the adapter may not have enough power available to run the 232:485 converter.*)

Screened twisted pair cable should be used. For longer cable runs or noisier environments, use of a cable specifically designed for RS485 may be necessary to achieve optimum performance. All "A" terminals should be connected together using one conductor of the twisted pair cable, all "B" terminals should be connected together using the other conductor in the pair. The cable screen should be connected to the "Gnd" terminals.

A Belden 9841 (Single pair) or 9842 (Two pair) or similar cable with a characteristic impedance of 120 ohms is recommended. The cable should be terminated at each end with a 120 ohm, quarter watt (or greater) resistor. Note: Diagram shows wiring topology only. Always follow terminal identification on Integra product label.

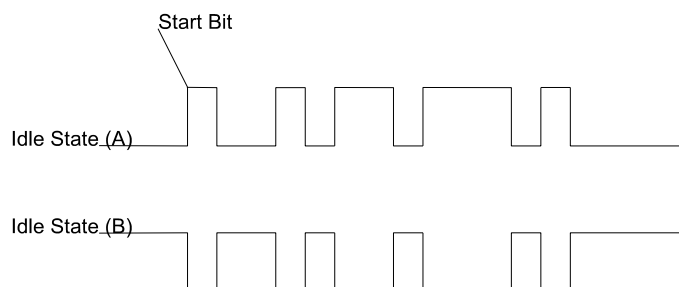


There must be no more than two wires connected to each terminal, this ensures that a "Daisy Chain" or "straight line" configuration is used. A "Star" or a network with "Stubs (Tees)" is not recommended as reflections within the cable may result in data corruption.



2.3 A and B terminals

The A and B connections to the Integra products can be identified by the signals present on them whilst there is activity on the RS485 bus:

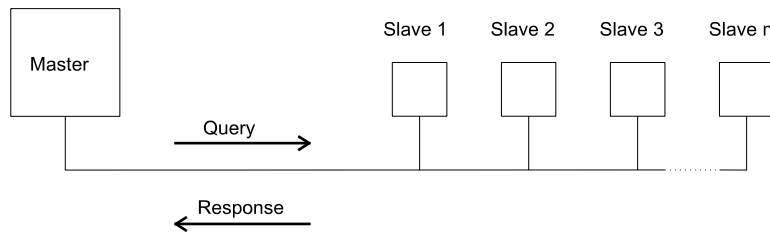


2.4 Troubleshooting

- Start with a simple network, one master and one slave. With Integra products this is easily achieved as the network can be left intact whilst individual instruments are disconnected by removing the RS485 connection from the rear of the instrument.
- Check that the network is connected together correctly. That is all of the “A’s” are connected together, and all of the “B’s” are connected together, and also that all of the “Gnd’s” are connected together.
- Confirm that the data “transmitted” onto the RS485 is not echoed back to the PC on the RS232 lines. (This facility is sometimes a link option within the converter). Many PC based packages seem to be not perform well when they receive an echo of the message they are transmitting. SpecView and PCView (PC software) with a RS232 to RS485 converter are believed to include this feature.
- Confirm that the Address of the instrument is the same as the “master” is expecting.
- If the “network” operates with one instrument but not more than one check that each instrument has a unique address.
- Each request for data must be restricted to 40 parameters (20 in the case of the older Integra 1000 or 2000) or less. Violating this requirement will impact the performance of the instrument and may result in a response time in excess of the specification.
- Check that the MODBUS mode (RTU or ASCII) and serial parameters (baud rate, number of data bits, number of stop bits and parity) are the same for all devices on the network.
- Check that the “master” is requesting floating-point variables (pairs of registers placed on floating point boundaries) and is not “splitting” floating point variables.
- Check that the floating-point byte order expected by the “master” is the same as that used by Integra products. (PCView and Citect packages can use a number of formats including that supported by Integra).
- If possible obtain a second RS232 to RS485 converter and connect it between the RS485 bus and an additional PC equipped with a software package, which can display the data on the bus. Check for the existence of valid requests.

3 MODBUS General Information

Communication on a MODBUS Network is initiated (started) by a “Master” sending a query to a “Slave”. The “Slave”, which is constantly monitoring the network for queries addressed to it, will respond by performing the requested action and sending a response back to the “Master”. Only the “Master” can initiate a query.



In the MODBUS protocol the master can address individual slaves, or, using a special “Broadcast” address, can initiate a broadcast message to all slaves. The Integra products do not support the broadcast address.

3.1 MODBUS Message Format

The MODBUS protocol defines the format for the master’s query and the slave’s response. The query contains the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field.

The response contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurred in receipt of the message then the message is ignored, if the slave is unable to perform the requested action, then it will construct an error message and send it as its response.

The MODBUS protocol functions used by the Integra products copy 16 bit register values between master and slaves. However, the data used by the Integra products is in 32 bit IEEE 754 floating point format. Thus each instrument parameter is conceptually held in two adjacent MODBUS registers.

Query

The following example illustrates a request for a single floating point parameter i.e. two 16-bit Modbus Registers.

First Byte								Last Byte
Slave Address	Function Code	Start Address (Hi)	Start Address (Lo)	Number of Points (Hi)	Number of Points (Lo)	Error Check (Lo)	Error Check (Hi)	

Slave Address: 8-bit value representing the slave being addressed (1 to 247), 0 is reserved for the broadcast address. The Integra products do not support the broadcast address.

Function Code: 8-bit value telling the addressed slave what action is to be performed. (3, 4, 8 or 16 are valid for Integra)

Start Address (Hi): The top (most significant) eight bits of a 16-bit number specifying the start address of the data being requested.

Start Address (Lo): The bottom (least significant) eight bits of a 16-bit number specifying the start address of the data being requested. As registers are used in pairs and start at zero, then this must be an even number.

Number of Points (Hi): The top (most significant) eight bits of a 16-bit number specifying the number of registers being requested.

Number of Points (Lo): The bottom (least significant) eight bits of a 16-bit number specifying the number of registers being requested. As registers are used in pairs, then this must be an even number.

Error Check (Lo): The bottom (least significant) eight bits of a 16-bit number representing the error check value.

Error Check (Hi): The top (most significant) eight bits of a 16-bit number representing the error check value.

Response

The example illustrates the normal response to a request for a single floating point parameter i.e. two 16-bit Modbus Registers.

First Byte									Last Byte
Slave Address	Function Code	Byte Count	First Register (Hi)	First Register (Lo)	Second Register (Hi)	Second Register (Lo)	Error Check (Lo)	Error Check (Hi)	

- Slave Address:** 8-bit value representing the address of slave that is responding.
- Function Code:** 8-bit value which, when a copy of the function code in the query, indicates that the slave recognised the query and has responded. (See also Exception Response).
- Byte Count:** 8-bit value indicating the number of data bytes contained within this response
- First Register (Hi)*:** The top (most significant) eight bits of a 16-bit number representing the first register requested in the query.
- First Register (Lo)*:** The bottom (least significant) eight bits of a 16-bit number representing the first register requested in the query.
- Second Register (Hi)*:** The top (most significant) eight bits of a 16-bit number representing the second register requested in the query.
- Second Register (Lo)*:** The bottom (least significant) eight bits of a 16-bit number representing the second register requested in the query.
- Error Check (Lo):** The bottom (least significant) eight bits of a 16-bit number representing the error check value.
- Error Check (Hi):** The top (most significant) eight bits of a 16-bit number representing the error check value.

* These four bytes together give the value of the floating point parameter requested.

Exception Response

If an error is detected in the content of the query (excluding parity errors and Error Check mismatch), then an error response (called an exception response), will be sent to the master. The exception response is identified by the function code being a copy of the query function code but with the most-significant bit set. The data contained in an exception response is a single byte error code.

First Byte		Last Byte		
Slave Address	Function Code	Error Code	Error Check (Lo)	Error Check (Hi)

- Slave Address:** 8-bit value representing the address of slave that is responding.
- Function Code:** 8 bit value which is the function code in the query OR'ed with 80 hex, indicating that the slave either does not recognise the query or could not carry out the action requested.
- Error Code:** 8-bit value indicating the nature of the exception detected. (See "Table Of Exception Codes" later).
- Error Check (Lo):** The bottom (least significant) eight bits of a 16-bit number representing the error check value.
- Error Check (Hi):** The top (most significant) eight bits of a 16-bit number representing the error check value.

3.2 Serial Transmission Modes

There are two MODBUS serial transmission modes, ASCII and RTU. Integra products do not support the ASCII mode.

In RTU (Remote Terminal Unit) mode, each 8-bit byte is used in the full binary range and is not limited to ASCII characters as in ASCII Mode. The greater data density allows better data throughput for the same baud rate, however each message must be transmitted in a continuous stream. This is very unlikely to be a problem for modern communications equipment. The format for each byte in RTU mode is:

Coding System:	Full 8-bit binary per byte. In this document, the value of each byte will be shown as two hexadecimal characters each in the range 0-9 or A-F.
Line Protocol:	1 start bit, followed by the 8 data bits. The 8 data bits are sent with least significant bit first.
User Option Of Parity And Stop Bits:	No Parity and 2 Stop Bits No Parity and 1 Stop Bit Even Parity and 1 Stop Bit. Odd Parity and 1 Stop Bit.
User Option of Baud Rate:	4800 ; 9600 ; 19200 ; 38400 (older Integra products do not support 38400 but do offer 2400 instead)

The baud rate, parity and stop bits must be selected to match the master’s settings.

3.3 MODBUS Message Timing (RTU Mode)

A MODBUS message has defined beginning and ending points. The receiving devices recognises the start of the message, reads the “Slave Address” to determine if they are being addressed and knowing when the message is completed they can use the Error Check bytes and parity bits to confirm the integrity of the message. If the Error Check or parity fails then the message is discarded.

In RTU mode, messages starts with a silent interval of at least 3.5 character times.

The first byte of a message is then transmitted, the device address.

Master and slave devices monitor the network continuously, including during the ‘silent’ intervals.

When the first byte (the address byte) is received, each device checks it to find out if it is the addressed device. If the device determines that it is the one being addressed it records the whole message and acts accordingly, if it is not being addressed it continues monitoring for the next message.

Following the last transmitted byte, a silent interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.

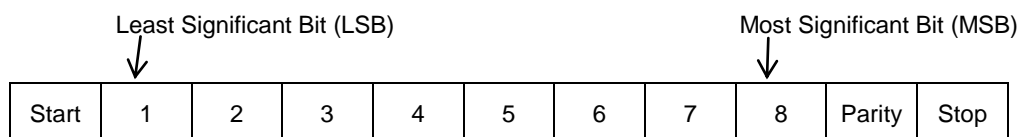
In the Integra 1000 and 2000, a silent interval of 60msec minimum is required in order to guarantee successful reception of the next request.

The entire message must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the message, the receiving device flushes the incomplete message and assumes that the next byte will be the address byte of a new message. Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving device may consider it a continuation of the previous message. This will result in an error, as the value in the final CRC field will not be valid for the combined messages.

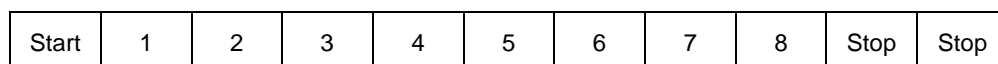
3.4 How Characters are Transmitted Serially

When messages are transmitted on standard MODBUS serial networks each byte is sent in this order (left to right):

Transmit Character = Start Bit + Data Byte + Parity Bit + 1 Stop Bit (11 bits total):



Transmit Character = Start Bit + Data Byte + 2 Stop Bits (11 bits total):

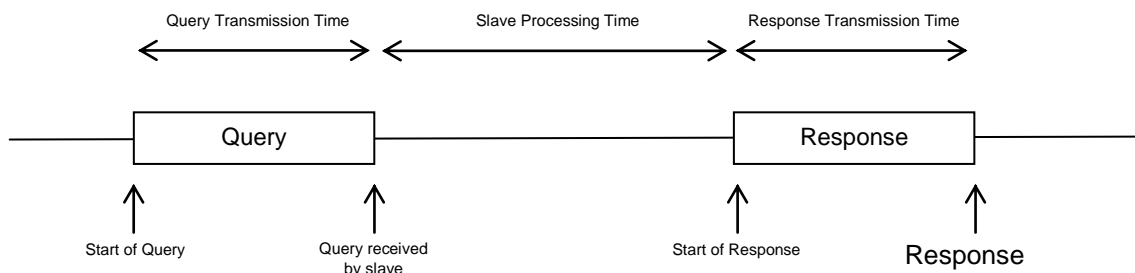


Integra products additionally support No parity, One stop bit.

Transmit Character = Start Bit + Data Byte + 1 Stop Bit (10 bits total):

Start	1	2	3	4	5	6	7	8	Stop
-------	---	---	---	---	---	---	---	---	------

The master is configured by the user to wait for a predetermined timeout interval. The master will wait for this period of time before deciding that the slave is not going to respond and that the transaction should be aborted. Care must be taken when determining the timeout period from both the master and the slaves' specifications. The slave may define the 'response time' as being the period from the receipt of the last bit of the query to the transmission of the first bit of the response. The master may define the 'response time' as period between transmitting the first bit of the query to the receipt of the last bit of the response. It can be seen that message transmission time, which is a function of the baud rate, must be included in the timeout calculation.



3.5 Error Checking Methods

Standard MODBUS serial networks use two error checking processes, the error check bytes mentioned above check message integrity whilst Parity checking (even or odd) can be applied to each byte in the message.

3.5.1 Parity Checking

If parity checking is enabled – by selecting either Even or Odd Parity - the quantity of “1’s” will be counted in the data portion of each transmit character. The parity bit will then be set to a 0 or 1 to result in an Even or Odd total of “1’s”.

Note that parity checking can only detect an error if an odd number of bits are picked up or dropped in a transmit character during transmission, if for example two 1’s are corrupted to 0’s the parity check will not find the error.

If No Parity checking is specified, no parity bit is transmitted and no parity check can be made. Also, if No Parity checking is specified and one stop bit is selected the transmit character is effectively shortened by one bit.

3.5.2 CRC Checking

The error check bytes of the MODBUS messages contain a Cyclical Redundancy Check (CRC) value that is used to check the content of the entire message. The error check bytes must always be present to comply with the MODBUS protocol, there is no option to disable it.

The error check bytes represent a 16-bit binary value, calculated by the transmitting device. The receiving device must recalculate the CRC during receipt of the message and compare the calculated value to the value received in the error check bytes. If the two values are not equal, the message should be discarded.

The error check calculation is started by first pre-loading a 16-bit register to all 1’s (i.e. Hex (FFFF)) each successive 8-bit byte of the message is applied to the current contents of the register. Note: only the eight bits of data in each transmit character are used for generating the CRC, start bits, stop bits and the parity bit, if one is used, are not included in the error check bytes. During generation of the error check bytes, each 8-bit message byte is exclusive OR’ed with the lower half of the 16 bit register. The register is then shifted eight times in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. After each shift the LSB prior to the shift is extracted and examined. If the LSB was a 1, the register is then exclusive OR’ed with a pre-set, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until all eight shifts have been performed. After the last shift, the next 8-bit message byte is exclusive OR'ed with the lower half of the 16 bit register, and the process repeated. The final contents of the register, after all the bytes of the message have been applied, is the error check value.

In the following pseudo code "ErrorWord" is a 16-bit value representing the error check values.

```

BEGIN
    ErrorWord = Hex (FFFF)
    FOR Each byte in message
        ErrorWord = ErrorWord XOR byte in message
        FOR Each bit in byte
            LSB = ErrorWord AND Hex (0001)
            IF LSB = 1 THEN ErrorWord = ErrorWord - 1
            ErrorWord = ErrorWord / 2
            IF LSB = 1 THEN ErrorWord = ErrorWord XOR Hex (A001)
        NEXT bit in byte
    NEXT Byte in message
END

```

3.6 Function Codes

The function code part of a MODBUS message defines the action to be taken by the slave. Integra products support the following function codes:

Code	MODBUS name	Description
03	Read Holding Registers	Read the contents of read/write location (4X references)
04	Read Input Registers	Read the contents of read only location (3X references)
08	Diagnostics	Only sub-function zero is supported. This returns the data element of the query unchanged.
16	Pre-set Multiple Registers	Set the contents of read/write location (4X references)

3.7 IEEE floating point format

The MODBUS protocol defines 16 bit "Registers" for the data variables. A 16-bit number would prove too restrictive, for energy parameters for example, as the maximum range of a 16-bit number is 65535. However, there are a number of approaches that have been adopted to overcome this restriction. Integra products use two consecutive registers to represent a floating-point number, effectively expanding the range to +/- 1x10³⁷.

The values produced by Integra products can be used directly without any requirement to "scale" the values, for example, the units for the voltage parameters are volts, the units for the power parameters are watts etc.

What is a floating point Number?

A floating-point number is a number with two parts, a mantissa and an exponent and is written in the form 1.234 x 10⁵. The mantissa (1.234 in this example) must have the decimal point moved to the right with the number of places determined by the exponent (5 places in this example) i.e. 1.234 x 10⁵ = 123400. If the exponent is negative the decimal point is moved to the left.

What is an IEEE 754 format floating-point number?

An IEEE 754 floating point number is the binary equivalent of the decimal floating-point number shown above. The major difference being that the most significant bit of the mantissa is always arranged to be 1 and is thus not needed in the representation of the number. The process by which the most significant bit is arranged to be 1 is called normalisation, the mantissa is thus referred to as a "normal mantissa". During normalisation the bits in the mantissa are shifted to the left whilst the exponent is decremented until the most significant bit of the mantissa is one. In the special case where the number is zero both mantissa and exponent are zero.

The bits in an IEEE 754 format have the following significance:

Data Hi Reg, Hi Byte.	Data Hi Reg, Lo Byte.	Data Lo Reg, Hi Byte.	Data Lo Reg, Lo Byte.
SEEE EEEE	EMMM MMMM	MMMM MMMM	MMMM MMMM

Where:

S represents the sign bit where 1 is negative and 0 is positive

E is the 8-bit exponent with an offset of 127 i.e. an exponent of zero is represented by 127, an exponent of 1 by 128 etc.

M is the 23-bit normal mantissa. The 24th bit is always 1 and, therefore, is not stored.

Using the above format the floating point number 240.5 is represented as 43708000 hex:

Data Hi Reg, Hi Byte	Data Hi Reg, Lo Byte	Data Lo Reg, Hi Byte	Data Lo Reg, Lo Byte
43	70	80	00

The following example demonstrates how to convert IEEE 754 floating-point numbers from their hexadecimal form to decimal form. For this example, we will use the value for 240.5 shown above. Note that the floating-point storage representation is not an intuitive format. To convert this value to decimal, the bits should be separated as specified in the floating-point number storage format table shown above. For example:

Data Hi Reg, Hi Byte	Data Hi Reg, Lo Byte	Data Lo Reg, Hi Byte	Data Lo Reg, Lo Byte
0100 0011	0111 0000	1000 0000	0000 0000

From this you can determine the following information.

- The sign bit is 0, indicating a positive number.
- The exponent value is 10000110 binary or 134 decimal. Subtracting 127 from 134 leaves 7, which is the actual exponent.
- The mantissa appears as the binary number 11100001000000000000000

There is an implied binary point at the left of the mantissa that is always preceded by a 1. This bit is not stored in the hexadecimal representation of the floating-point number. Adding 1 and the binary point to the beginning of the mantissa gives the following:

1.111000010000000000000000

Now, we adjust the mantissa for the exponent. A negative exponent moves the binary point to the left. A positive exponent moves the binary point to the right. Because the exponent is 7, the mantissa is adjusted as follows:

11110000.1000000000000000

Finally, we have a binary floating-point number. Binary bits that are to the left of the binary point represent the power of two corresponding to their position. For example, 11110000 represents $(1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 240$.

Binary bits that are to the right of the binary point also represent a power of 2 corresponding to their position. As the digits are to the right of the binary point the powers are negative. For example: .100 represents $(1 \times 2^{-1}) + (0 \times 2^{-2}) + (0 \times 2^{-3}) + \dots$ which equals 0.5.

Adding these two numbers together and making reference to the sign bit produces the number +240.5.

For each floating point value requested two MODBUS registers (four bytes) must be requested. The received order and significance of these four bytes for Integra products¹ is shown below:

Data Hi Reg, Hi Byte	Data Hi Reg, Lo Byte	Data Lo Reg, Hi Byte	Data Lo Reg, Lo Byte
-------------------------	-------------------------	-------------------------	-------------------------

3.8 MODBUS Commands supported

All Integra products support the “Read Input Register” (3X registers), the “Read Holding Register” (4X registers) and the “Pre-set Multiple Registers” (write 4X registers) commands of the MODBUS RTU protocol. All values stored and returned are in floating point format to IEEE 754 with the most significant register first².

3.8.1 Read Input Registers

MODBUS code 04 reads the contents of the 3X registers.

Example

The following query will request ‘Volts 1’ from an instrument with node address 1:

Field Name	Example (Hex)
Slave Address	01
Function	04
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	02
Error Check Low	71
Error Check High	CB

Note: Data must be requested in register pairs i.e. the “Starting Address” and the “Number of Points” must be even numbers to request a floating point variable. If the “Starting Address” or the “Number of points” is odd then the query will fall in the middle of a floating point variable the product will return an error message.

The following response returns the contents of Volts 1 as 230.2. But see also “Exception Response” later.

¹ The Integra 1500/1600 Series supports the ability to reverse the register order. In reverse register mode the registers are transmitted and received Data Lo Register followed by Data Hi Register. Note, the Data Hi Register is the one that contains the sign bit.

² Except where the instrument is an Integra 1500 Series configured for reversed register mode.

Field Name	Example (Hex)
Slave Address	01
Function	04
Byte Count	04
Data, High Reg, High Byte	43
Data, High Reg, Low Byte	66
Data, Low Reg, High Byte	33
Data, Low Reg, Low Byte	34
Error Check Low	1B
Error Check High	38

3.8.2 Read Holding Registers

MODBUS code 03 reads the contents of the 4X registers.

Example

The following query will request the prevailing 'Demand Time':

Field Name	Example (Hex)
Slave Address	01
Function	03
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	02
Error Check Low	C4
Error Check High	0B

Note: Data must be requested in register pairs i.e. the "Starting Address" and the "Number of Points" must be even numbers to request a floating point variable. If the "Starting Address" or the "Number of points" is odd then the query will fall in the middle of a floating point variable the product will return an error message.

The following response returns the contents of Demand Time as 1, But see also "Exception Response" later.

Field Name	Example (Hex)
Slave Address	01
Function	03
Byte Count	04
Data, High Reg, High Byte	3F
Data, High Reg, Low Byte	80
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	F7
Error Check High	CF

3.8.3 Write Holding Registers

MODBUS code 10 (16 decimal) writes the contents of the 4X registers.

Example

The following query will set the Demand Time to zero, which effectively resets the Demand Period:

Field Name	Example (Hex)
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	00
Number of Registers High	00
Number of Registers Low	02
Byte Count	04
Data, High Reg, High Byte	00
Data, High Reg, Low Byte	00
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	F2
Error Check High	AF

Note: Data must be written in register pairs i.e. the “Starting Address“ and the “Number of Points” must be even numbers to write a floating point variable. If the “Starting Address” or the “Number of points” is odd then the query will fall in the middle of a floating point variable the product will return an error message. In general only one floating point value can be written per query

The following response indicates that the write has been successful. But see also “Exception Response” later.

Field Name	Example (Hex)
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	00
Number of Registers High	00
Number of Registers Low	02
Error Check Low	41
Error Check High	C8

3.9 Exception Response

If the slave in the “Write Holding Register” example above, did not support that function then it would have replied with an Exception Response as shown below. The exception function code is the original function code from the query with the MSB set i.e. it has had 80 hex logically ORed with it. The exception code indicates the reason for the exception. The slave will not respond at all if there is an error with the parity or CRC of the query. However, if the slave can not process the query then it will respond with an exception. In this case a code 01, the requested function is not support by this slave.

Field Name	Example (Hex)
Slave Address	01
Function	10 OR 80 = 90
Exception Code	01
Error Check Low	8D
Error Check High	C0

3.9.1 Table of Exception Codes

Integra products support the following function codes:

Exception Code	MODBUS name	Description
01	Illegal Function	The function code is not supported by the product
02	Illegal Data Address	Attempt to access an invalid address or an attempt to read or write part of a floating point value
03	Illegal Data Value	Attempt to set a floating point variable to an invalid value
05	Slave Device Failure	An error occurred when the instrument attempted to store an update to it's configuration

3.10 Diagnostics

MODBUS code 08 provides a number of diagnostic sub-functions. Only the "Return Query Data" sub-function (sub-function 0) is supported on Integra 1540, 1560, 1580 and 1630.

Example

The following query will send a diagnostic "return query data" query with the data elements set to Hex(AA) and Hex(55) and will expect these to be returned in the response:

Field Name	Example (Hex)
Slave Address	01
Function	08
Sub-Function High	00
Sub-Function Low	00
Data Byte 1	AA
Data Byte 2	55
Error Check Low	5E
Error Check High	94

Note: Exactly one register of data (two bytes) must be sent with this function.

The following response indicates the correct reply to the query, i.e. the same bytes as the query.

Field Name	Example (Hex)
Slave Address	01
Function	08
Sub-Function High	00
Sub-Function Low	00
Data Byte 1	AA
Data Byte 2	55
Error Check Low	5E
Error Check High	94

4 Modbustm TCP (Ethernet)

INTEGRA 1630 options include an Ethernet communication module for connection to SCADA systems using the MODBUS TCP protocol. The INTEGRA with Ethernet option module acts as a MODBUS slave device and may be queried by a MODBUS master device. All messages sent to the INTEGRA Ethernet interface must conform to the MODBUS TCP protocol. For details see MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b Downloadable from the Modbus-IDA, www.Modbus-ida.org

The Integra Ethernet option module supports 10/100Base-T Ethernet communication. Connection is via an Ethernet switch that supports the IEEE 802.3 standard at 10/100Mbps. The Integra is fitted with a socket suitable for an RJ45 connector. Use a CAT5 or CAT6 patch cord to connect the meter to an Ethernet switch or hub. A suitable cable is available from Tyco Netconnect : CAT6 5m LSZH patch cord, p/n 0-1711093-5. Alternatively, for permanent installations, connect to the meter using a suitable installed network cable.

The MODBUS TCP protocol is used for data exchange between HMI/SCADA applications and the INTEGRA. The network architecture must include a MODBUS TCP client, (PC). TCP/IP port 502 is reserved for MODBUS messages.

Data Coding and format is identical to Modbus RTU as described in section 1 of this document. Set Modbus slave address to 1 in Modbus TCP master software.

4.1 Communication Parameters

The front panel of the INTEGRA provides access to the set up sequence of the meter. In the set up sequence it is possible to modify the settings for baud rate, parity and slave ID. The communication parameters of an Integra fitted with an Ethernet option module refer to internal communications within the meter, **do not modify these settings** or the Ethernet interface will cease to function.

For reference these values are factory set to:

Modbus Address: 001
Baud rate: 38.4kBaud
Parity: no parity 1 stop bit

4.2 IP Address Assignment

The IP address of the Integra must be unique and appropriate for the network to which it is attached. The address to use will depend upon the local network and should be determined by the network administrator. The Integra Ethernet option module supports static IP address assignment only.

The Integra IP address is factory set to "192.168.1.100". If attaching two or more Integra meters to the same network the IP addresses must be changed so that each meter is assigned to an unique address.

4.2.1 Connections for configuring the IP address

Preferably, set the IP address using a direct point to point connection between the PC and Integra.

If this is not practical, for example, if replacing an Integra in an existing network that has suffered accidental damage, it is possible to set the IP address via the Ethernet network, provided that no other device on the network already uses the Integra factory default address.

To connect the Integra to a PC directly an Ethernet crossover patch cable (Cat 5 UTP) is required.

Some Ethernet adapters will auto configure transmit and receive lines. If the PC to be used does not do this then a crossover lead is required. In general, a standard Ethernet cable will not suffice. Standard leads are used between Ethernet nodes and a bridge or switch when wiring a conventional network with many Ethernet nodes.

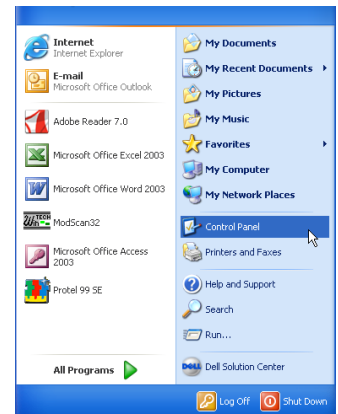
4.2.2 Configuring a PC for Ethernet Integra

Ideally, set aside a PC specially for configuring Integra as described. If this is not practical, and there also is a need to connect the PC to the organisation network, **ensure, in advance, that your network administrator is fully familiar with the intended procedure as described below. It is the user's responsibility to ensure that any local IT policies are complied with.**

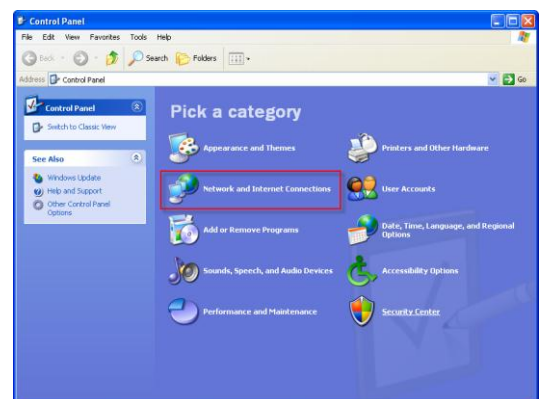
The PC to be used will need a copy of the Ruinet utility installed. (download from www.crompton-instruments.com)

To enable the PC to communicate with the Integra the local area network settings for the PC must be set to appropriate values, with the first three number groups the same as the Integra IP address. The settings are made using the Windows "Control Panel" utility application. If the PC is normally used on the site wide network then disconnect the PC from that network before making the changes described.

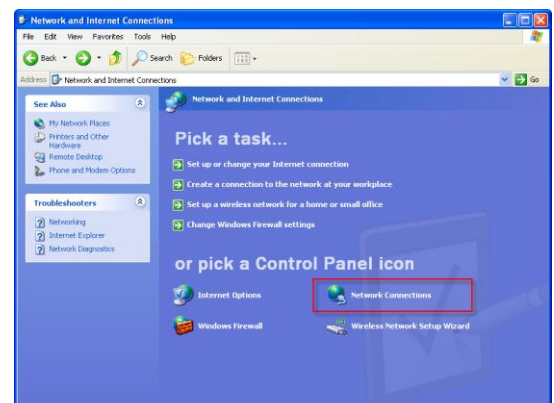
Launch the Windows Control Panel from the Windows "Start" menu. This example shows Windows XP. Other versions of Windows will require a similar process but the details and screens may differ.



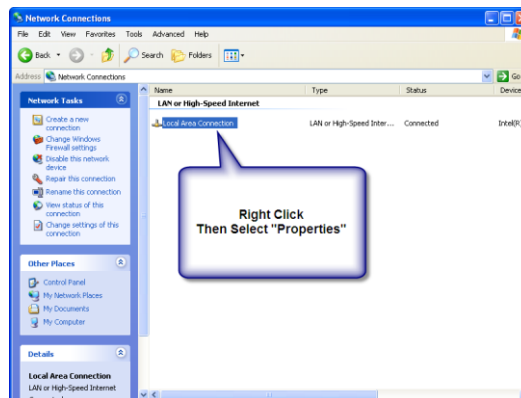
In the "Control Panel" window: click on the "Network and Internet Connections" item. This will activate the "Network and Internet Connections" window.



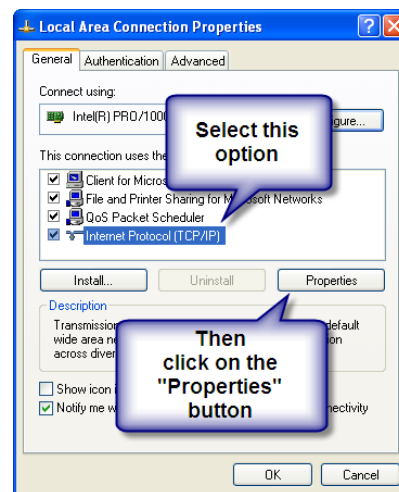
In the "Network and Internet Connections" window: click on the "Network Connections" item, this will activate the "Network Connections" window.



Right click on the “Local Area Connection” item and select “Properties” in the popup window that appears. This will activate the “Local Area Connection Properties” window.



In the “This connection uses the following items:” section select the “Internet Protocol (TCP/IP)” item and click on the “Properties” button. This will activate the “Internet Protocol (TCP/IP) Properties” window.



Select the “Use the following IP address” option and set the IP address and subnet mask as described below. **Before making any changes, carefully note the previous settings – they may be essential to re-establishing the PC on the organisation network. Be sure to revise all settings to previous values before attempting to reconnect the target PC to the organisation network.** If in doubt, consult your organisation’s network administrator.

The IP address shown in this example is suitable to connect to an Integra set to the default IP address, as the default address starts with 192.168.1....

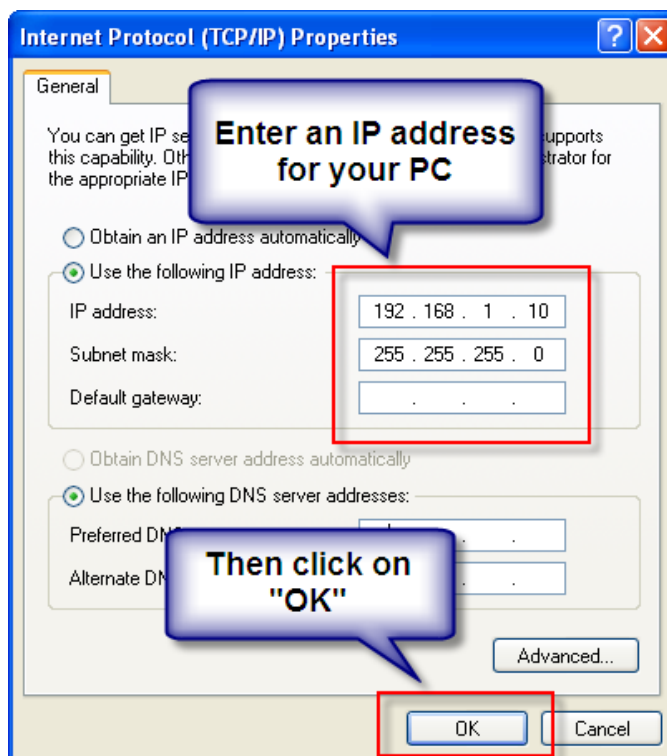
For example, if the test meter is assigned to the IP address “192.168.1.100”, then a suitable IP address for the PC is “192.168.1.nnn”, where n can be any value, (apart from 100, as this is already used by the Integra). In this example we have used 10. Enter the subnet mask as shown above. If the Integra does not have the factory default IP address, then set the first three number groups of the PC IP address to the same as those for the target Integra.

Click on “OK” to close the window.

Click on “OK” to close the “Local Area Connection Properties” window.

Close the “Network Connections” window.

The PC is now ready to communicate with the Integra.



Connect the patch cable to the RJ45 connector on the Integra and plug the other end of the cable into the network port of the PC.

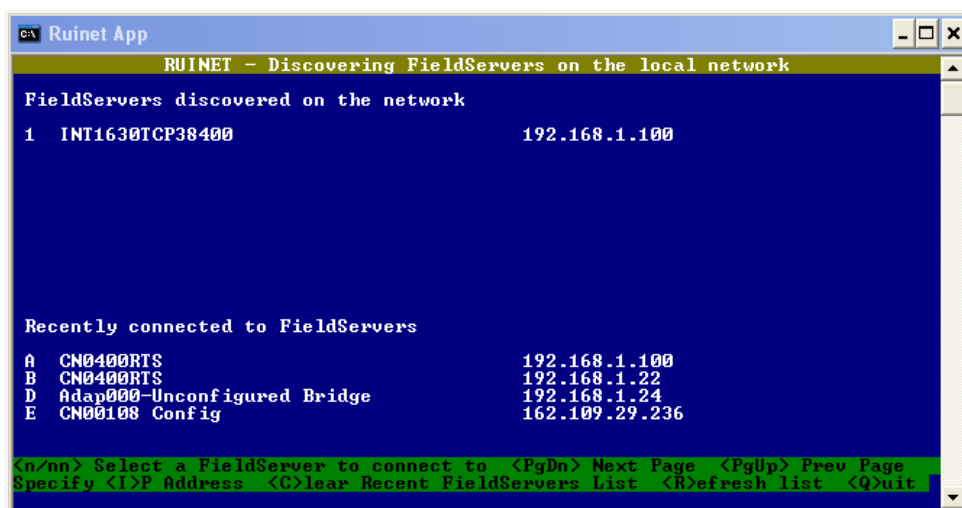
Warning: Ensure that any wiring to the Integra is unpowered whilst making an Ethernet connection and that there is no access to the Integra terminals or associated wiring whilst it is powered. Ensure the Ethernet cable is positioned so that it cannot accidentally touch any live wiring.

Launch the "RUInet.exe" application by double clicking on the name from Windows Explorer. The application will find the Integra meter, (and any other similar devices on the MODBUS[®]TCP network).

The "Ruiet.exe" application will search for all appropriate devices on the network. The initial screen depends on the devices it finds. (It may also prompt you to disable any firewall, in which case simply press the "1" key).

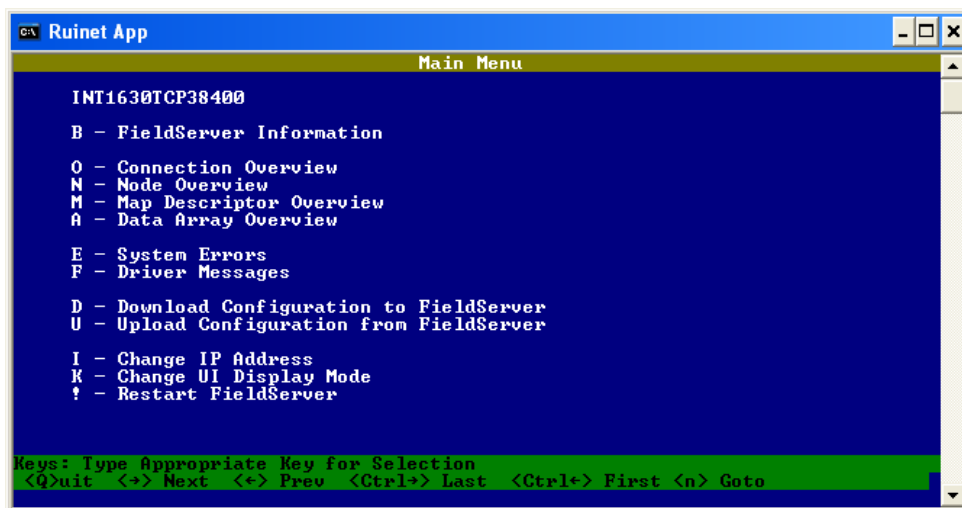
Ruiet will either display the "Discovering" window, as shown, or the "Main Menu", in which case skip the first step.

If the screen shown appears, the top list, "... discovered on the network" will typically have only one entry when making a point to point connection from the PC to the Integra. If an attempt is made to configure an Integra which is already connected to a larger network, then multiple entries may be shown. If multiple units are shown, the new Integra can be identified because it will have the default IP address (192.168.1.100). Press the number key next to the correct entry in the list, ("1" in our example).

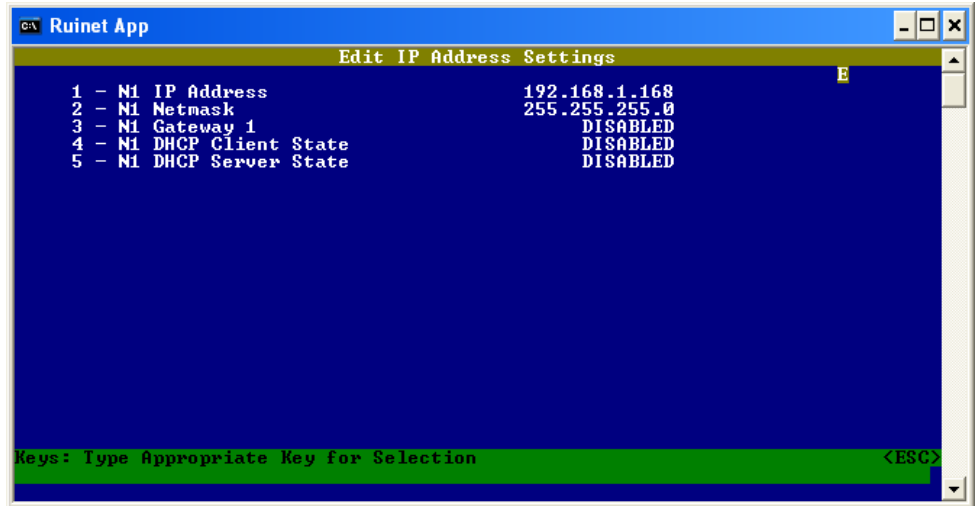


The Ruinet application is now addressing the Integra to be configured and the main menu will appear.

Select "Change IP Address by pressing "I"."

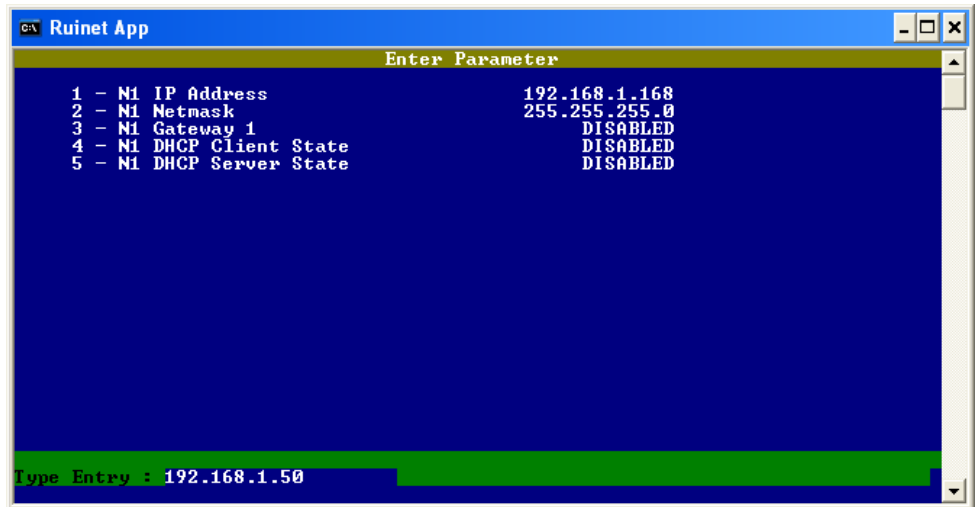


Select "1 – N1 IP Address" in the "Edit IP Address Settings" window... by pressing "1"



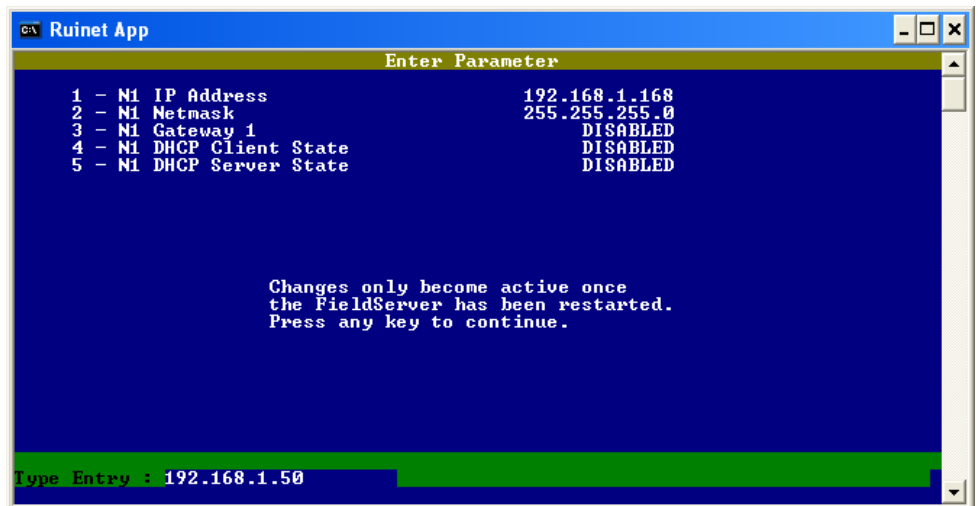
The lower bar will prompt for entry of the new IP address. Type the IP address that the Integra will use in future and then press the <Enter> key.

Note: The Ruinet application may not display the correct value for the current IP address. This does not affect the operation.



If Ruinet is able to change the IP address this screen appears. Press a key to complete the process. There will be a delay of a few seconds then the "edit IP address" screen will return.

If required, select option 2 and change the N1 Netmask. The default netmask is 255.255.255.0. Only use this option when a different net mask is required.



When all changes are complete, press the <ESC> key to exit the IP address settings window and then press the "Q" key to exit the ruinet application.

Power cycle the Integra to make the new settings active.

4.3 Single Register Response

Some SCADA systems scan their network for active devices by sending a read request for one register to each node address in turn, expecting either a data value or an exception code. In either case the SCADA system can record that node as being active.

The Integra 1630 with Modbus TCP protocol option will not respond with exception codes. For compatibility with these SCADA systems the Integra 1630 will respond to any single register read request with the same hardcoded value of "1630". This applies to both input and holding registers and for addresses 1 to 9999.

5 BACnet IP interface

5.1 Introduction

INTEGRA 1630 options include an Ethernet communication module for connection to SCADA systems using the BACnet IP protocol. The Integra 1630 acts as a server device and waits to receive commands from a BACnet/IP client. A BACnet/IP client (e.g. a SCADA system running on a PC), is used to instigate communication with the meter. All messages sent to the INTEGRA Ethernet interface must conform to the BACnet IP protocol, within the command subset defined below. For details on the protocol see the BACnet organisation website: <http://www.bacnet.org/>

The Integra Ethernet option module supports 10/100Base-T Ethernet communication. Connection is via an Ethernet switch that supports the IEEE 802.3 standard at 10/100Mbps. The Integra is fitted with a socket suitable for an RJ45 connector. Use a CAT5 or CAT6 patch cord to connect the meter to an Ethernet switch or hub. A suitable cable is available from Tyco Netconnect : CAT6 5m LSZH patch cord, p/n 0-1711093-5. Alternatively, for permanent installations, connect to the meter using a suitable installed network cable.

Data Coding and format is based heavily on the Modbus RTU format as described in section 1 of this document. The BACnet/IP interface is configured to give the fastest possible response to queries for all analogue parameter values.

5.2 Communication Parameters

The front panel of the INTEGRA provides access to the set up sequence of the meter. In the set up sequence it is possible to modify the settings for baud rate, parity and slave ID. The communication parameters of an Integra fitted with an Ethernet option module refer to internal communications within the meter. **Do not modify these settings** or the Ethernet interface will cease to function.

For reference these values are factory set to:

Modbus Address: 001
Baud rate: 38.4kBaud
Parity: no parity 1 stop bit

5.3 IP Address Assignment

The IP address of the Integra must be unique and appropriate for the network to which it is attached. The address to use will depend upon the local network and should be determined by the network administrator. The Integra Ethernet option module supports static IP address assignment only.

The Integra IP address is factory set to "192.168.1.100". If attaching two or more Integra meters to the same network the IP addresses must be changed so that each meter is assigned to a unique address. General instructions for changing the IP address are shown in section 4.2, however, it will be necessary to continue that process to ensure that the following three items match the intended network.

- 1 - N1 IP Address
- 2 - N1 Netmask
- 3 - N1 Gateway 1 (Must be in same sub-net as IP Address)

It is important to ensure all three items are set correctly. Make the settings effective by resetting or power cycling the instrument.

5.4 Initialisation and register identification.

When it receives a "WHOIS" message the Integra returns an identification message, the "IAM" message, listing its BACnet node ID. The BACnet client device generates a table of the BACnet devices on the network mapping the node IDs to the IP address of each device. Each device is addressed using its node ID, thus imposing the restriction that node IDs must be unique. The user does not require knowledge of the IP address, as the node ID is sufficient.

The Integra is designed to ensure that each unit in a manufacturing batch has a unique node ID, but the user may also influence that node ID if for example, there is a Node ID clash with other, previously installed BACnet devices. The user can change the node ID from factory default as described in section 5.7.

Once the client device has built its network table it is possible to start communicating with the Integra. The client system requires information as to which queries the Integra supports and the meaning of each return value. This information is available on the Integra 1630 PICS sheet, and is shown below, or the client may gather the information from the Integra itself, using a BACnet ReadObject command. This returns the instance number of each supported object, (register), in the Integra.

The object table of the Integra is split into 2 sections, the first section lists all the “Analogue Values” of the Integra. Analogue Values may be read or write, these are analogous to the Integra Modbus holding registers. Each object (register) is assigned an instance number, equal to the Integra parameter number.

The second section lists all the “Analogue Inputs” of the Integra. Analogue Inputs are read only - these are analogous to the Integra input registers. Each object (register) is assigned an instance number, equating to the Modbus parameter number.

5.5 Supported Queries

This guide only includes BACnet/IP query types which are supported by the Integra 1630. The only relevant query types are those that read values from or write values to the Integra.

To read a parameter from the Integra a “ReadProperty” query is required, with the object type set to “Analogue Value”. The instance number is set to the appropriate parameter number and the property identifier set to “Present Value”. Sending the above query to the Integra, results in a response from the Integra giving the most recent calculated value of the queried parameter.

To write to an Integra register a “WriteProperty” query is required, with the object type set to “Analogue Value”. The instance number set to the appropriate parameter number and the property identifier set to “Present Value”. Sending the above query to the Integra; sets the value of the queried parameter.

BACnet systems should not attempt to address parameters whose instance value is not defined. Some parameters are reserved for factory use and selecting these may give unpredictable results.

ReadPropertyMultiple query is supported by the Integra, but only to the extent of simultaneously reading single values and getting the associated quality value from the Integra, (The quality value indicates that the value is current, as defined by the timeout value factory set at 1.5 seconds. The ReadPropertyMultiple query may not be used to return an array of data points and for this reason the PICS sheet does not specify support for the ReadPropertyMultiple query

5.6 Protocol Implementation Conformance Statement

Date	April 10th 2008
Vendor Name	Tyco Electronics Energy Division
Product Name	Integra 1630 Digital Metering System
Product Model Number(s)	INT 1630-X-X-X-080
Firmware Version	2.26
BACnet Protocol Revision	2

Product Description

The Integra 1630 is a multi function digital metering instrument offering measurement, display and communication of many electrical parameters. The Integra 1630 is programmable via a simple menu driven interface and can be integrated into BACnet IP systems.

BACnet Standardized Device Profile (Annex L): BACnet Application Specific Controller (B-ASC)

BACnet Interoperability Building Blocks Supported (Annex K): DS-RP-A, DS-WP-A, DM-DDB-B, DM-DOB-B, DM-DCC-B

Segmentation Capability: Segmentation not supported

Standard Object Types Supported: No dynamic Creation or Deletion supported. No proprietary properties or object types

Device Object:

Optional Properties Supported:	Description
Writable Properties:	None
Property Range Restrictions:	na

Analogue Input Object:
 Optional Properties Supported: Description
 Writable Properties: None
 Property Range Restrictions: na

Analogue Value Object:
 Optional Properties Supported: Description
 Writable Properties: Present_Value
 Property Range Restrictions: See Table

Analogue Value Ranges – for valid ranges see section 1.3. All analogue values correspond directly to their Modbus equivalents except for AV200 and AV201, node ID values, which are exclusive to the BACnet interface.

Table 1: Analogue Value Objects

Item	Parameter	Equates to Modbus register
AV1	Demand Time	40001
AV8	Energy Reset	40015
AV13	Password	40025
AV50	Hours Run Reset	40099
AV200	* Node ID Offset	40469
AV201	* Node ID Value	40471

* See section 5.7.

Data Link Layer Options: BACnet IP, (Annex J)
 Device Address Binding: Static device binding is not supported. (No client functionality is included)
 Networking Options: None
 Character Sets Supported: ANSI X3.4
 Device Node ID: Units are shipped with a pseudo-random node ID based on an internal serial number. To change the ID see section 5.7.

Table 2: BACnet Analogue Input Objects

Analogue input object numbering is shown in the table for Modbus input registers in section 1.2. By way of example the first few entries are shown below, along with the device object.

BACnet Object	Parameter	Units
Device Object	This Device. Integra 1630 Digital Metering System with BACnet/IP auxiliary. Returns the value Integra_1630_BCN	N/A
Analogue Input Object 1	Volts 1	VOLTS
Analogue Input Object 2	Volts 2	VOLTS
Analogue Input Object 3	Volts 3	VOLTS
Analogue Input Object 4	Current 1	AMPS
	For other values, consult table in section 1.2	

5.7 Changing the BACnet Node ID

This procedure applies to instruments manufactured from the start of 2009. Older instruments may have the node ID configured by the procedure set out in the notes following this section.

The BACnet node IDs for these instruments are generated automatically from the Integra serial number and will typically result in every BACnet Integra having a unique node ID as delivered. However, there is a degree of user control in case of a clash with another node on the target network.

By default the lower four digits of the second serial number are taken as the BACnet node ID. If required the user can prefix these four digits with one digit in the range 1 to 5. Alternatively, the user can select a node ID in the range 0 to 249.

The node ID options described above are controlled by the node ID offset, AV 200 (Analogue Value 200), as shown in the following table.

Table 3: Node ID Offset To 16 Bit Node ID Conversion Table

Node Offset (AV 200)	Node ID (AV 201)
0	0
1	1
2	2
to	to
247	247
248	248
249	249
Default 250	NNNN
251	1NNNN
252	2NNNN
253	3NNNN
254	4NNNN
255	5NNNN

Note:

Node Offset, AV 200, is a 'write only' object so will not necessarily show the current value of the node offset.

Node Number, AV 201, is a 'read only' object.

NNNN are the low four digits of the second serial number, AV 23.

Read node ID object, AV 201, to see the current node ID. If the node ID needs to be changed then proceed as follows.

Note: The node ID offset is password protected to minimise the possibility of inadvertent alteration, so the steps below include the unlocking process.

First read the password object, AV 13. If it shows a value of 1 then the password has already been entered. If it shows a value of 0 then write the password to AV 13. If a password has not been setup then use the default value of 0.

The password object, AV 13, should now read 1.

Next, select the required Node Offset from the table above then write that value to the node offset object, AV 200.

Check the node ID setting by reading the node ID value object, AV 201.

The new node ID will take effect after the instrument is power cycled.

When configuring instruments with unique numbers it may be helpful to mark the selected number on the outside of the instrument.

Changing the BACnet Node ID on older instruments

These notes only apply to products delivered up to the end of 2008. The BACnet node ID is set to 11 during manufacture. The node ID can be changed by the user, however to do so requires editing the configuration file inside the BACnet interface, where, any change other than to the node ID could prevent the interface from operating. It is recommended that changing the node ID be carried-out while the instrument is connected one-to-one to a PC during initial set-up before it is connected to a live BACnet network.

The procedure for changing the node ID requires the use of three software tools, 'ruinet.exe' the network utility for the BACnet interface, MS 'notepad.exe' or similar text editor and a BACnet test utility. Connect the instrument to the PC using an Ethernet crossover cable.

Warning: Ensure that any wiring to the Integra is unpowered whilst making an Ethernet connection and that there is no access to the Integra terminals or associated wiring whilst it is powered. Ensure the Ethernet cable is positioned so that it cannot accidentally touch any live wiring.

Note, the network settings of the PC must be compatible with those of the instrument. After a minute or so with the PC and instrument powered-up it will be possible to detect the instruments BACnet interface using ruinet. Once detected the instrument can be selected using a number key. From the main menu for the instrument, press the U key twice to upload the current configuration file from the instrument to the PC. The actual file will be called config.csv and will be placed in the same directory as ruinet. At this point it will be a good idea to save a copy of config.csv in another directory. The file config.csv in the ruinet directory will now need to be edited, using the text editor, to change both occurrences of the old node ID to the new node ID. The node ID occurs in the 'Common Information' section and 'Server Side Nodes' section. Save the edited file using the original file name. The Integra 1630 allows node IDs in the range 0 to 65,535. From the main menu for the instrument, in ruinet, press the D key twice to download the file config.csv to the instrument. If a BACnet test utility is available it is recommended this be used to verify the new node ID before connecting the modified instrument to the live BACnet network.

5.8 Procedure For Changing The Node-Name

The instrument node-name, the name available to the BACnet network, is set during manufacture, to 'Integra_1630_BCN'. This name can be changed by the user, however to do so requires editing the configuration file inside the BACnet interface, where, any change other than to the node-name could prevent the interface from operating. It is recommended that changing the node-name be carried-out while the instrument is connected one-to-one to a PC during initial set-up before it is connected to a live BACnet network.

Warning: Ensure that any wiring to the Integra is unpowered whilst making an Ethernet connection and that there is no access to the Integra terminals or associated wiring whilst it is powered. Ensure the Ethernet cable is positioned so that it cannot accidentally touch any live wiring.

The procedure for changing the node-name requires the use of two software tools, 'ruinet.exe' the network utility for the BACnet interface and MS 'notepad.exe' or similar text editor. Connect the instrument to the PC using an Ethernet crossover cable. Note: the network settings of the PC must be compatible with those of the instrument. After a minute or so with the PC and instrument powered-up it will be possible to detect the instruments BACnet interface using ruinet. Once detected the instrument can be selected with a number key then the node-name displayed on the node page by pressing the N key. Press the escape key to get to the main menu for the instrument, then press the U key twice to upload the current configuration file from the instrument to the PC. The actual file will be called config.csv and will be placed in the same directory as ruinet. At this point it will be a good idea to save a copy of config.csv in another directory. The file config.csv in the ruinet directory will now need to be edited, using the text editor, to change every occurrence of 'Integra_1630_BCN' to the new node-name. There are 70 occurrences of the node-name so it is best to use a search and replace feature of the text editor. Save the edited file using the original file name. See below for recommendations for selecting node-names. From the main menu for the instrument, in ruinet, press the D key twice to download the file config.csv to the instrument. When the download is done, power cycle the instrument and check the new node-name as above.

Node-names can be from 1 to 32 characters long, however, we recommend the following rules for selecting node-names.

- 1) Node-names should be from 2 to 24 characters long.
- 2) Node-names should start with an alpha character, upper or lower case.
- 3) Remaining Node-name characters can be upper or lower case alphas, numbers or an underscore character.

Examples of legitimate node names include: M1, DMS_Cold_Room_03, Integra_1630_BCN_147, etc.

6 RS485 Implementation of Johnson Controls Metasys

These notes briefly explain Metasys and Crompton Instruments Integra integration. Use these notes with the Metasys Technical Manual, which provides information on installing and commissioning Metasys N2 Vendor devices.

6.1 Application details

The Integra is an N2 Vendor device that connects directly with the Metasys N2 Bus. This implementation assigns key electrical parameters to ADF points, each with override capability.

Components requirements

- Integra with RS485 card and N2 port available.
- N2 Bus cable.

6.1.1 Metasys release requirements

- Metasys Software Release 12.04 or later
- NCM-361-8 or Metasys Extended Architecture NAE35,NAE45,NAE55

Integra may be compatible with earlier releases of N2 software, but Johnson Controls only supports integration issues on the above.

6.1.2 Support for Metasys Integration

Primary Contact: Mr Martin Langman
Service Support Manager
Unit 1 ,Kestrel Rd
Trafford Park M17 1SF
England
Email: martin.langman@jci.com
Direct Tel +44 (0) 161 848 6674
Fax +44 (0) 161 848 7196

6.1.3 Support for Crompton Integra operation

This is available via local sales and service centre.

6.1.4 Design considerations

When integrating the Crompton equipment into a Metasys Network, keep the following considerations in mind.

- Make sure all Crompton equipment is set up, started and running properly before attempting to integrate with the Metasys Network.
- A maximum of 32 devices can be connected to any one NCM N2 Bus segment, or up to 100 devices if repeaters are used.

Device Address	1-247 (Limited by co-resident Modbus™ protocol)
Port Set-up:	
Baud Rate*	9600
Duplex	Half
Word Length	8
Stop Bits*	1
Parity*	None
Interface	RS485

* The user should ensure these values are set as shown on the Integra for compatibility with the N2 network.

6.2 METASYS N2 Integra Point Mapping table

ADF Point	Parameter Description	Units
1	Voltage 1	V
2	Voltage 2	V
3	Voltage 3	V
4	Current 1	A
5	Current 2	A
6	Current 3	A
7	Voltage Average	V
8	Current Average	A
9	Power Sum	kW
10	VA Sum	kVA
11	var Sum	kvar
12	Power Factor Average	
13	Frequency	Hz
14	Active Energy Import (6 digits max)	kWh
15	Reactive Energy Import (6 digits max)	kvarh
16	Watts Demand Import	kW
17	Maximum Watts Demand Import	kW
18	Amps Demand	A
19	Maximum Amps Demand	A
20	Voltage L1 - L2	V
21	Voltage L2 - L3	V
22	Voltage L3 - L1	V
23	Neutral Current	A
24	Active Energy Import (less ADF 14)	GWh
25	Reactive Energy Import (less ADF 15)	Gvarh
26	THD V1	%
27	THD V2	%
28	THD V3	%
29	THD I1	%
30	THD I2	%
31	THD I3	%
32	THD Vmean	%
33	THD Imean	%
34	Power 1	kW
35	Power 2	kW
36	Power 3	kW
37	VA 1	kVA
38	VA 2	kVA
39	VA 3	kVA
40	var 1	kvar
41	var 2	kvar
42	var 3	kvar
43	PF 1	
44	PF 2	
45	PF 3	
46	PA 1	Degrees
47	PA 2	Degrees
48	PA 3	Degrees
49	A Sum	A
50	VLL Average	V

51	Export Wh (6 digits max)	kWh
52	Export Wh (less ADF 51)	GWh
53	Export varh (6 digits max)	kvarh
54	Export varh (less ADF 53)	Gvarh
55	Command Register	See note below.
56	VAh (6 digits max)	kVAh
57	VAh (less ADF 56)	GVAh
58	VA Demand	A
59	Maximum VA Demand	A
60	I1 Demand	A
61	Maximum I1 Demand	A
62	I2 Demand	A
63	Maximum I2 Demand	A
64	I3 Demand	A
65	Maximum I3 Demand	A
66	PA Average	Degrees
67	Hours Run (6 digits max)	mh
68	Hours Run (less ADF 67)	kh

Command Register: The command register, ADF point 55, is used to reset groups of accumulated values held in the instrument. A reset is performed by overriding the Command Register with the values shown in the following table. It is not essential to release the Command Register for the reset to operate.

To reset,	Override the Command Register with,
all energy values.	156001.
all demand values.	156002.
hours run count.	156003.

Energy and Hours Run: Energy and hours run values are available from pairs of ADF points – one showing the least significant 6 digits and the other showing the overflow, or more significant digits. The total value since the last reset is found by combining the value from both of these points appropriately. This allows energy and hours run values rollover to be postponed, and maintains consistency.

The ADF point pairs are:

- 14 and 24 (active import energy)
- 15 and 25 (reactive import energy)
- 51 and 52 (active export energy)
- 53 and 54 (reactive export energy)
- 56 and 57 (apparent energy)
- 67 and 68 (hours run)

7 Integra Profibus Interface

The Integra Profibus interface implements a “modular” slave, in which the I/O set is not fixed but allows modules, real or virtual, to be selected (“plugged in”) at configuration time. The configuration tool will be specific to the manufacturer of the Profibus network Class-1 Master unit. However, the slave device GSD file provides the data to the configuration tool that will associate a module type (e.g. “VA Phase 1”) with configuration data.

7.1 GSD file

A copy of the GSD file may be accessed via the website : www.crompton-instruments.com, on the same page as Integra instruction and operation manuals.

7.2 Floating Point Format

As with Modbus, data transfer occurs using IEEE floating point format.

The convention for the four bytes of floating-point data is to transmit and receive it in Big Endian format, that is, the most significant byte is in Data0 and the least significant byte is in Data3. More details on this format are shown in section 3.7

7.3 Single Parameter access

The system described doesn’t allow access to all the Integra parameters simultaneously, as only 50 parameters are available in the configuration table for the Integra. Since the configuration is fixed in operation, modules can’t be swapped in and out dynamically to gain access to addition parameters. So in addition to the regular parameter modules, a ‘Control’ module has been defined. Through the use of its I/O area a single parameter read or write facility is available.

Note: that only one instance of a ‘Control’ module can be used in an Integra configuration and the module must be placed after other configured modules in the table.

In order to reference the parameters, the Modbus numbering convention is used. These are shown in section 1.3.

The table below illustrates the contents of the I/O area of the Control module.

Output Bytes	Input Bytes
Command	Echo command
Modbus address hi	Echo address hi
Modbus address lo	Echo address lo
Data0 write	Data0 read
Data1 write	Data1 read
Data2 write	Data2 read
Data 3 write	Data3 read
Status	Status/error

7.4 Functionality of the PLC Function Block

The PLC programmer will need to implement a function block that can fulfil the following requirements.

7.4.1 Reading

In order to execute a read of parameter values from the Control module of the Integra, the sequence of events is as follows:

The Command should initially be null (i.e. zero). The Modbus Address is written into the output fields, then the ‘Read floating-point value from an input register’ (the value 4), or read floating-point value from a holding register’ (the value 3) is written to the command field.

The slave checks for a change in the command field as indication that there is a command to action and that the rest of the fields are valid and ready for use.

The slave will return the appropriate value in the data field, together with a confirmation/rejection value in the status field. The master checks for the command field to be echoed in the input data as indication that the rest of the fields are valid, and ready for reading. A zero in the status field indicates the command completed correctly. A non-zero value indicates an error has occurred.

The sequence is completed by outputting the null-command (zero) and awaiting confirmation (zero) from the echo-command byte.

This last step is important since Profibus transmits the I/O data continuously and repeatedly, so the slave device looks for the change in the command field to initiate the action. If this step is omitted, even though the data field may have been changed the slave will take no action unless it detects a change in the command field.

7.4.2 Writing

In order to write parameter values using the Control module, the sequence of events is as follows:

The Command should initially be null (i.e. zero). The Modbus Address is written into the output fields, then the value to be written is placed in the data field. The next item to be put into the data area is the command 'Write floating point value to Holding Register' (the value 16), written to the command field.

The slave checks for a change in the command field as indication that there is a command to action and that the rest of the fields are valid and ready for use.

The slave will return the appropriate value in the data field, together with a confirmation/rejection value in the status field. The master checks for the command field to be echoed in the input data as indication that the rest of the fields are valid, and ready for reading. A zero in the status field indicates the command completed correctly. A non-zero value indicates an error has occurred.

The sequence is completed by outputting the null-command (zero) and awaiting confirmation (zero) from the echo-command byte.

This last step is important since Profibus transmits the I/O data continuously and repeatedly, so the slave device looks for the change in the command field to initiate the action. If this step is omitted, even though the data field may have been changed the slave will take no action unless it detects a change in the command field.

7.5 Common Problems

The most common reasons why a single parameter read or write fails are:

The Command field contains a value not supported by the module. (Values 0, 3, 4, and 16 are permissible).
The Modbus Address is incorrect. Correct Modbus address values can be found in the table of available parameters contained in this guide.

A parameter is being written to that requires the password to be entered in the password register before it can be changed.

7.6 Modules available

Values which may be returned directly via modules are shown below:

Volts1 (L1-N 4W or L1-L2 3W)
Volts2 (L2-N 4W or L2-L3 3W)
Volts3 (L3-N 4W or L3-L1 3W)
Current 1
Current 2
Current 3
W Phase 1
W Phase 2
W Phase 3
VA Phase 1
VA Phase 2
VA Phase 3

var Phase 1
var Phase 2
var Phase 3
Power Factor Phase 1
Power Factor Phase 2
Power Factor Phase 3
Phase Angle Phase 1
Phase Angle Phase 2
Phase Angle Phase 3
Volts Ave
Current Ave
Current Sum
Watts Sum
VA Sum
var Sum
Power Factor Ave
Average Phase Angle
Frequency
Wh Import
Wh Export
varh Import
varh Export
W Demand Import
W Max Demand Import
A Demand
A Max Demand
V L1-L2 (calculated)
V L2-L3 (calculated)
V L3-L1 (calculated)
Average Line to Line Volts
Neutral Current
THD Volts 1
THD Volts 2
THD Volts 3
THD Current 1
THD Current 2
THD Current 3
THD Voltage Mean
THD Current Mean
Power Factor (+Ind/-Cap)