Cutler-Hammer

Modbus MINT Application

Application Note

Description — Page

General Description

This application note will describe some of the details in applying the Modbus MINT (mMINT) with the IQ Transfer.

mMINT Programming

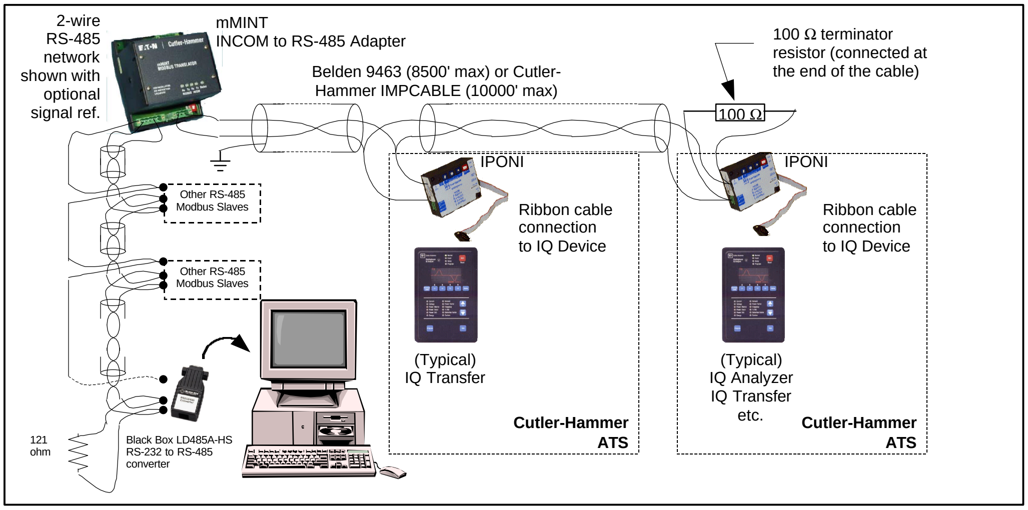
Refer to Instruction Leaflet IL 66A7508H01 for more details.

The mMINT can be used immediately without external programming as long as the mMINT default register assignments are acceptable.

This application note will describe a test set up using the Black Box LD485A-HS connected to an mMINT. The mMINT is connected via IMPCABLE to an IPONI on an IQ Transfer.

In this application note, the Modbus master role is handled by a personal computer running the Modscan (www.win-tech.com) software. Modscan emulates a Modbus master on a MS Windows PC. You may use this program or write your own. Modbus is a very well supported protocol and many third party drivers and toolkits are available. See www.modbus.org for more information on the Modbus protocol as well as links to shareware and JAVA open source code files.

ATC 400/600/800 Direct Connect-Modbus MINT (mMINT)



Modbus MINT (mMINT)
The mMINT connects an RS-485 based Modbus master device (e.g. computer serial port port, modem, PLC) to the INCOM network. The mMINT supports bit rates of 1200, 9600 and 19200 bps. Supports INCOM data rate of 9600 bits per second only.

Since the mMINT has an RS-485 port and our experiment has a computer with only an RS-232 port. We must install an RS-232 to RS-485 converter.

RS-485 to RS-232 Converter
The RS-485 signal is suitable for transmission for up to 4000 feet using Belden 3106A or equivalent low capacitance cable. Since most computers do not include RS-485 ports, this test used a converter (Black Box LD485A-HS) to convert the 2-wire twisted pair to a 25-pin RS-232 port.

A conventional "AT modem" cable connected the computer to the LD485A-HS. See detailed drawing later in this document for the pinout connections for this cable.

The mMINT RS-485 port utilize the third wire "signal reference" lead defined by the RS-485 standard. The LD485A-HS does not include a terminal for that signal. However that signal is available on an edge connector on the LD-485 board. See detailed drawing for instructions on connecting (soldering) a wire to that gold edge and bringing that wire to a terminal block for connection to the RS-485 network.

Lab testing showed that if the LD485 and mMINT were connected to the same AC L1 (hot), L2 (neutral) and ground and the cable distance between the two devices was only a few feet, the 3rd signal common lead was not needed.

Note that unlike INCOM or Lonworks, RS-485 is a polarity sensitive standard. When multidropping cable between nodes, connect each terminal marked "non-inverting (+)" together and connect all terminals marked "inverting (-)" together in a daisy chain fashion.

Note that the mMINT and the LD485A-HS have both "non-inverting" and "inverting" terminals. Unfortunately, neither device refers to them by that name, instead they refer to them by the letters "A" and "B". Even more unfortunate is the mMINT and LD485A-HS associate different functions to these letters. See table below:

	mMINT	LD485A
non-inverting	A	B
inverting	B	A

Therefore, to connect the mMINT to the LD485A, you must connect the mMINT "A" to the LD485A-HS "B" terminals (both TxB and RxB). Likewise you must connect the mMINT "B" terminal to the LD485A-HS "A" terminals (both TxA and RxA). See detailed drawing later in this document for connections used for this test.

Refer to the mMINT instruction leaflet IL 66A7508H01 for more information.

If you intend to connect other devices to this RS-485 multi-drop network, carefully examine the function of each RS-485 terminal of each device to determine whether the signal is the "non-inverting (+)" or the "inverting (-)". Connect these terminals to the like terminals of the other devices.

Belden 9463 or IMPCABLE
To minimize communications problems, use this specified twisted shielded pair between the mMINT and the IPONI(s). The INCOM cabling standard permits taps from the main cable up to 200 feet long. Each tap may have up to 64 nodes. See Cutler-Hammer document TD17513 (http://www.ch.cutler-hammer.com/unsecure/cms1/00017513.PDF) for more information. Note that the shield connection is grounded only at the mMINT. All other shield connnections are tied together but left ungrounded.

IPONI
The INCOM Product Operated Network Interface translates the low level data signals obtained from the IQ device into robust signal that can be transmitted for 8500 feet using Belden 9463 or up to 10000 feet using IMPCABLE.

In this test we set the IPONI to address 001.

100 Ohm Terminator Resistor
Minimizes reflections by absorbing excess energy that has been transmitted onto the twisted pair cable by INCOM devices. 1/4 watt, carbon, 5% recommended. Do not use wirewound resistors.

The mMINT includes a DIP switch that can be switched on to close a built-in 100 ohm termination resistor, eliminating the need to add one at the mMINT. Note: if the mMINT is not at the end of the INCOM cabling run, do not close the termination resistor or add a termination resistor. Termination resistors are only added at the end of the INCOM cable (beween the furthest two points if the cable is not wired as a bus).

121 (or 150) Ohm Terminator Resistor
RS-485 networks typically have a higher impedance and so require a different termination resistance. As with the INCOM termination resistor, use a carbon type, 5%, 1/4 watt (or larger) device. Do not use wirewound resistors as they do not present a "pure" resistance, but rather contain a significant reactance component at the frequencies that appear on this network. The result is that the reflections are not attenuated as effectively as with carbon resistors.

Ordering Instructions
Order the transfer switch with option 48A to include the IPONI mounted. Order the mMINT by description from your transfer switch representative.

Also needed is the Belden 9463 or CH IMPCABLE cable for the INCOM wiring and Belden 3106A or equivalent for the RS-485 wiring and the Black Box LD485A-HS RS-485 to RS-232 converter (the Black Box C620A-F will work also, but details of that device are not discussed in this applicatoin note).

Finally some type of Modbus master is required. To set up a test system, install the ModScan (www.win-tech.com) software on a computer and connect an "AT modem cable" from the computer COM port to the LD485A-HS as shown on the detailed drawing later in this document.

ModScan Software
From the Setup / Serial menu select the COM port that is connected to the LD485A-HS. Set the baud to match the

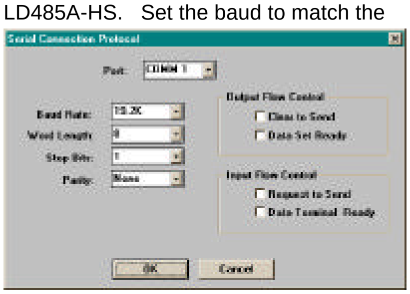


Figure 1: Modscan menu accessed from Setup / Serial pull down menu.

DIP switch settings on the mMINT (this test tried all three available settings -- 1200, 9600 and 19200 bps, all worked well). Set the data bits to 8 and parity to none. Make sure all hardware handshaking selections are not checked (listed as "Output Flow Control" and "Input Flow Control" on the Setup / Serial pull down menu).

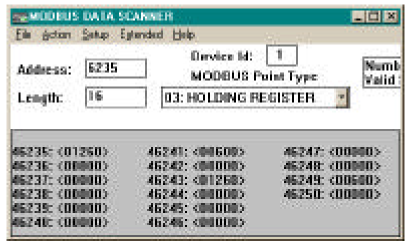


Figure 4: Action / Start Poll has been selected and data is being read from the mMINT.

Enter these values into the main screen:
Address: 6235
Length: 16
Device ID: 1
03: Holding Register

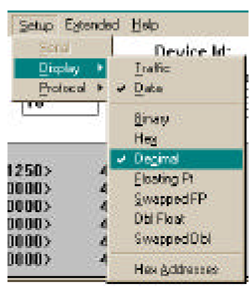


Figure 3: Select Extended / Display / Decimal to display the returned data in the decimal form

Select Action / Start Poll to begin polling the network. Most likely one or more problems will occur and you will receive a flashing message stating "Modbus Poll Status Timeout". This indicates that a message was transmitted but no response was heard.

Note that while you can double-click on a register displayed in Modscan, this method cannot be used to change a value within the mMINT. The reason is that this method utilizes the Modbus function code 6 (Preset Register) rather than the function code 16 (Preset Multiple Registers). The mMINT only supports changing registers with function code 16.

Modscan does include the function code 16 changing from a pull down menu

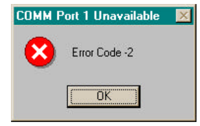


Figure 2: If you receive this message when attempting to communicate with the mMINT it indicates that another Windows program is also using the same serial port. Make sure that programs such as Palm Pilot Hotsync managers or other programs running in the background have been exited. Windows operating system does not allow two programs to access the same serial port at the same time.

Extended / Preset Registers, however the test version of Modscan would not support register numbers greater than 409999 (offset 1 through 9999). This may be fixed in later versions of Modscan.

Even if your version of Modscan does not support writing to a register greater than 409999, you can still write using the Extended / User Defined pull down menu.

To use the User Defined function, you must understand the format of a Modbus message as the User Defined function requires that you load the individual bytes to be transmitted to form a complete Modbus message.

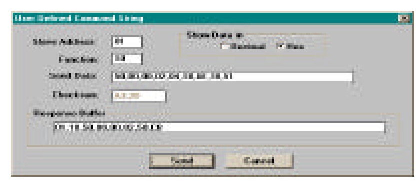


Figure 3: Example of User Defined function screen. Accessed from Extended / User Defined pull down menu. You may chose either decimal or hex format to enter data.

Function 16: Preset Multiple Registers

Example: Write $186E_{16}$ to register 420481 (5000_{16}) and 1851_{16} to register 420482 (5001_{16}) both to Modbus node 001 (01_{16}).

Query Transmitted:

Byte	Example	Explanation
1	01_{16}	Slave address
2	10_{16}	Function
3	50_{16}	Address Hi
4	00_{16}	Address Lo
5	00_{16}	No. of Regs Hi
6	02_{16}	No. of Regs Lo
7	04_{16}	Byte Count

Byte	Example	Explanation
8	18_{16}	Data Word 1 Hi
9	$6E_{16}$	Data Word 1 Lo
10	18_{16}	Data Word 2 Hi
11	51_{16}	Data Word 2 Lo
12	$A3_{16}$	CRC (calculated)
13	$2D_{16}$	CRC (calculated)

Response Returned

Byte	Example	Function
1	01_{16}	Slave address
2	10_{16}	Function
3	50_{16}	Address Hi
4	00_{16}	Address Lo
5	00_{16}	No. of Regs Hi
6	02_{16}	No. of Regs Lo
7	50_{16}	CRC
8	$C8_{16}$	CRC

Ignore Bad Addresses within Blocks

Modbus allows up to 125 registers to be retrieved with one message. However, when a block is read that includes a range of invalid addresses bordered by supported addresses, the mMINT responds with an exception response 02 Invalid Register Address. This means that you must break up your request into multiple messages that only request valid

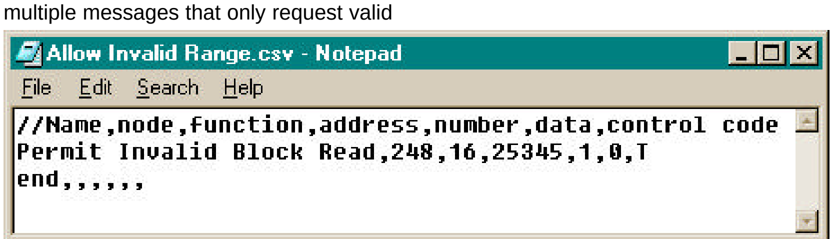


Figure 4: Script to write a "0" to mMINT register 425345. When this is executed (Extended / Run Script), the mMINT will not longer respond to a read to an invalid address or range of addresses an exception 02 "Illegal address". Instead invalid registers are returned with the data fields padded with zeros.

registers. To configure the mMINT to not send this exception when a reading a block of registers that may contain one or more ranges of invalid (not available) registers, set register 405325 (6300_{16}) of address 248 (mMINTs address) equal to 0. If this register is written (using function 16 "Preset Multiple Registers") to a zero, the mMINT will not transmit an exception 02 when a read request covers a range of invalid registers. In that case, the mMINT will pad the response with 0's in the location assigned to the invalid addresses. Note that you must use function code 16 not function code 6 to write that register. Also, note that you must have the Diagnostic DIP switch set to Enable (open) to activate the mMINT's ability to respond to a message intended for Modbus node 248.

Creating a Custom Table

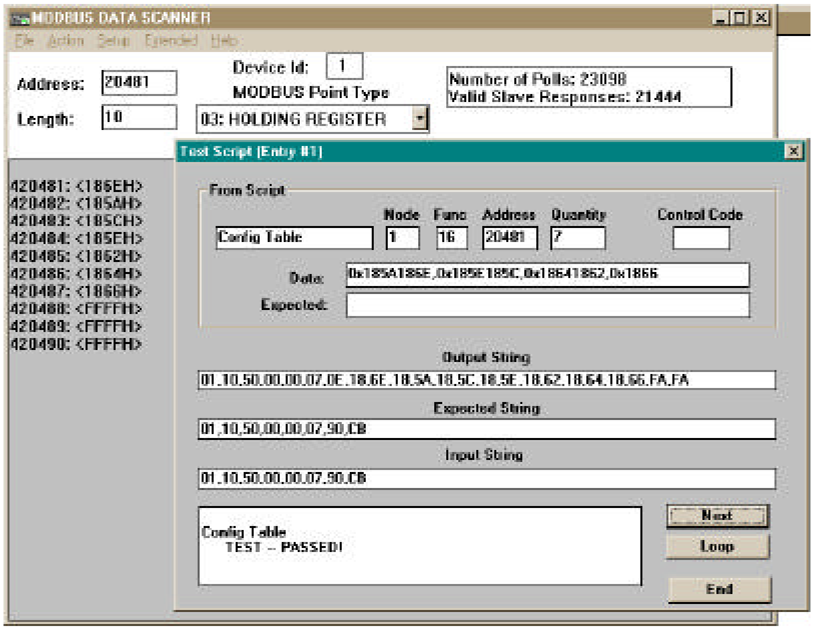


Figure 10: Running the mMINT.CSV script loads registers 420481 as shown above. Note that that each data value is a 32-bit value ordered as <HighWord Low Word>.

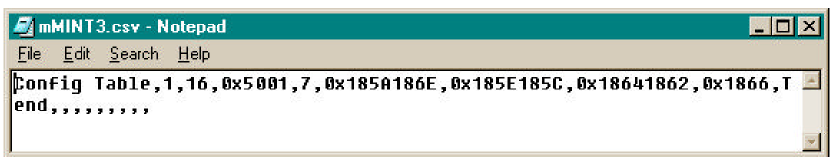


Figure 11: Contents of the mMINT3.CSV script that sets the registers 420481 as shown above.

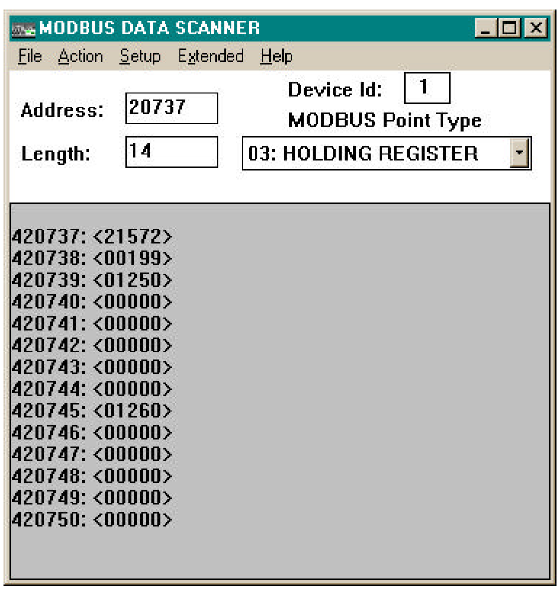


Figure 12: Results of loading custom table. Notice that 420737 contains PROD ID, 420738 contains STATUS, 420738 contains the low order word of Vab (high order word in 420740 = 0), etc.

The mMINT permits you to reorder the data registers where data is stored within the mMINT. This way you can retrieve data from the IQ Transfer in a register order of your chosing. This is accomplished by writing into a table a group of values that corresponds to the type of data requested. The order that you place these values into the table dictates how the data is read from the table.

This is best explained by example. Suppose that we wished to place the PROD ID (406255), the SOURCE 1 voltages (406235 - 406242) and the SOURCE 2 (406243 - 406250) voltages of IQ Transfer, INCOM address 001 into a block of registers starting with 420737 (5100_{16}).

To accomplish this, we must write a configuration table beginning at register 420481 (5000_{16}) as shown:

Node	Register	Write	Meaning
001	420481	$186E_{16}$	PROD ID
001	420482	1851_{16}	S1 Vab
001	420483	$185C_{16}$	S1 Vbc
001	420484	1860_{16}	S1 Vbc
001	420485	1862_{16}	S2 Vab
001	420486	1864_{16}	S2 Vbc
001	420487	1866_{16}	S2 Vca

This table is written to address 001, not address 248.

The results of loading this table is that a table in memory (always starting at 420737) will now contain data as shown:

Register	Meaning
420737	PROD ID
420738	STATUS
420739	S1 Vab (least sig. word)
420740	S1 Vab (most sig. word)
420741	S1 Vbc (least sig. word)
420742	S1 Vbc (most sig. word)
420743	S1 Vca (least sig. word)
420744	S1 Vca (most sig. word)
420745	S2 Vab (least sig. word)
420746	S2 Vab (most sig. word)
420747	S2 Vbc (least sig. word)
420748	S2 Vbc (most sig. word)
420749	S2 Vca (least sig. word)
420750	S2 Vca (most sig. word)

Troubleshooting

No LEDs are flashing on the LD485 and no LEDs (other than the Status LED) are flashing on the mMINT

- Check to that Modscan is set to the proper COM port.
- Check that the LD485A DCE/DTE jumper is set to DCE
- Check that the LD485A is powered
- Verify the cable connecting the computer COM port to the LD485A matches the drawing "mMINT Wiring Diagram" found later in this document

LEDs are flashing on the LD485 and the Rx LED is flashing, but the Tx LED isn't flashing

- Check that the TxA and RxA terminals on the LD485A are connected to the "B" terminal on the mMINT and check that the TxB and the RxB terminals on the LD485A are connected to the "A" terminal on the mMINT.
- Verify that the bit rate set in the computer matches the bit rate on the mMINT DIP switches
- Make sure the terminal blocks on the mMINT are screwed down tightly, both the wire holding screws as wells a the screws that mechanically hold the pull out portion of the terminal securely in tis base.

Modscan flashes "Illegal Data"

- Verify that the registers being read are valid addresses.
- If reading a double register, make sure that Modbus is configured to read both. If you attempt to read only one register, the mMINT responds with the exception code 03 "Illegal Data".

Intermittant errors causing retries on the INCOM network

- Insure that the 100 ohm terminating resistors are installed at the ends fo the network.
- Insure that the wiring is seated tightly under the terminals on the mMINT and the IPONI.

Default Register Assignment - IQ Transfer

Register	Bit	Description
406145	15–8	See Table 9 "Primary Status Code Definitions (See mMINT Instruction Leaflet IL 66A7508H01 for Tables 9, 10 and 11)
	7–0	See Table 10 Secondary Status Code Definitions
406146		See Table 11 Cause of Status Codes
406255		Product ID
406256	15–8	All zeros
	7–6	Bit # 7 6: 0 0 On good source (see bits 2, 5); 0 1 Calling for engine start; 1 0 Transferred (see bits 2, 5); 1 1 In alarm
	5	1= Source 2 is connected to load
	4	1= IQ Transfer has been power cycled since last read of 406256
	3	1= New unread time-stamped transfer (history) buffer available
	2	1= Source 1 is connected to load
	1	1= Source 2 is available
	0	1= Source 1 is available

Bit # 7	6	
0	0	On good source (see bits 2, 5)
0	1	Calling for engine start
1	0	Transferred (see bits 2, 5)
1	1	In alarm

Register				
406235	Vab	Source 1	(x 10)	(least significant word)
406236	Vab	Source 1	(x 10)	(most significant word)
406237	Vbc	Source 1	(x 10)	(least significant word)
406238	Vbc	Source 1	(x 10)	(most significant word)
406239	Vca	Source 1	(x 10)	(least significant word)
406240	Vca	Source 1	(x 10)	(most significant word)
406241	Hz	Source 1	(x 10)	(least significant word)
406242	Hz	Source 1	(x 10)	(most significant word)
406243	Vab	Source 2	(x 10)	(least significant word)
406244	Vab	Source 2	(x 10)	(most significant word)
406245	Vbc	Source 2	(x 10)	(least significant word)
406246	Vbc	Source 2	(x 10)	(most significant word)
406247	Vca	Source 2	(x 10)	(least significant word)
406248	Vca	Source 2	(x 10)	(most significant word)
406249	Hz	Source 2	(x 10)	(least significant word)
406250	Hz	Source 2	(x 10)	(most significant word)

Note: Value = (65536*[most significant word])+[least significant word]

Note: (x10) means the number is scaled by 10.
True Value = Value / 10

Note: Tables referenced are found in the mMINT Instruction Leaflet IL66A7508H01

Testing Comments

1. When requesting any voltage or Hz, remember that this is a two word pair.
   If you only request a single register, the mMINT will return Modbus exception 03 "Invalid Data".
2. If you request registers that are not available, mMINT will return exception 03 "Invalid Data"
3. When requesting the voltages and Hz beginning with 406235, you must ask for a length in a multiple of two. In other words, if you want to read voltages and Hz from both sources, request a length of 16 (406235 through 406251). A request for an odd number of registers will return an exception 03 "Invalid Data" from the mMINT.
4. Section 6.3 describes how to create a custom table of registers. This feature is used if you wish to group into a continuous series of registers data that may have been scattered around multiple registers within the mMINT. Since the IQ Transfer only has 3 separate tables (406145 - 406146, 406255 - 406256 and 406235 - 406251), this feature may be of limited interest. However, note that if you do not program this custom table (beginning at 420481), any attempt to read from that table will return an exception code 03 "Illegal Data".
5. *** IMPORTANT ***
   The mMINT's RS-485 port is a 2-wire port with terminals marked "A", "B", "COM" and "SLD".
   "A" is defined by the mMINT as the "non-inverting" (+) input
   "B" is defined by the mMINT as the "inverting" (-) input
   THIS IS TYPICALLY DIFFERENT THAN INDUSTRY DE FACTO USAGE. Most RS-485 devices will reverse those two meaning. Consequently, when connecting the mMINT to an RS-485 pay particular attention to insure that all "non-inverting" inputs are connected common to each other and all "inverting" inputs are connected to each other. Refer to the Cutler-Hammer diagram for using the Black Box LD-485A-HS RS-232 to RS-485 converter. Notice that this device connects the "A" terminal on the Black Box to the "B" terminal on the mMINT and vice versa.
6. Many RS-485 to RS-232 converters do not bring the COM terminal to an external terminal. COM is used as a signal reference between each RS-485 device and is used to maintain a common ground reference. RS-485 devices can operate without a COM wire (3rd wire in the 2-wire interface), but will stop communicating if the RS-485 A or B wire line-to-ground voltage at one end of the network is 10 volts or more different than the RS-485 A or B wire to ground at the other end of the wire. Refer to *mMINT Interface Diagram* for details on how to wire to the mMINT COM terminal from the Black Box LD-485A-HS.

Changing Preferred Source

The mMINT permits a “pass-through” control mode whereby any INCOM command can be transmitted. To demonstrate the power of this command, we choose an extremely complex example.

In this example we send a message that tells the transfer switch to select Source 2 as the preferred source. If Source 2 is a generator, this will cause the IQ Transfer to issue an engine start command. Once the generator voltage and frequency are within tolerance, the IQ Transfer will switch the load to Source 2. This can be useful if you wish to remotely dispatch a generator.

A much simplier method of issuing a remote engine start is to issue a command to perform a remote engine load test. Refer to documentation provided in the Windows Help menu of the Eaton Dialer program (ED) for differences in remotely dispatching using an engine test.

Step 1 : Send Modbus message 1: Write "Read Setpoints 3 C 9" message to mMINT.
Step 2 : Send Modbus message 2: Read returned setpoint data.
Step 3 : Change Preferred Source Selector byte, re-compute checksum and write data back to buffer.
Step 4 : Send Modbus message 3: Write "Download Setpoints 3 F 9" message to mMINT.
Step 5 : Send Modbus message 4: Write message 1 of 3 F 9
Step 6 : Send Modbus message 5: Write message 2 of 3 F 9
Steps 7 - 47 : Send Modbus messages 6 through 46 : Write messages 3 through 43 of 3 F 9.

IQ Transfer: Function (3 C 9) Transmit (Read) Setpoints Buffer
Modbus MINT Cross-Reference

Operation: Upon receipt of 3 C 9, the IQ Transfer returns a block of 43 messages, 24 bits long divided into 3 bytes. The Modbus MINT loads these 3 bytes into two registers as shown.

Send this Modbus register pair first. This transmits the 3 C 9 command (INST = 3, COMM = C, SCOMM = 9)

Number		Hi Byte Contains	Lo Byte Contains	Register	High Byte	Low Byte
1	==>	Byte 0	Control	424577	1100 0011 = C3h = 243 decimal	1000 0001 = 81h = 129 decimal
1	==>	Byte 2	Byte 1	424578	1001 0000 = 90h = 144 decimal	0000 0001 = 01h = 1 decimal

Byte 0
bits 0 - 3 : INST field (3)
bits 4 - 7 : COMM field (C)

Control
bits 0 - 5 : number of expected response messages from device
(in this example we expect 1 message [ACK] as reply)
bit 6 : 0 (reserved)
bit 7 : 0 - data message
1 - control messages

Byte 2
bits 0 - 3 : bits 8 - 11 of address of device (*0000* 0000 0001b)
bits 4 - 7 : SCOMM field (9)

Byte 1
bits 0 - 7 : bits 0 - 7 of address of device (0000 *0000 0001*b)

In response to this message sent, the following Modbus registers are filled with the reply (setpoint buffer received from the IQ Transfer):

	INCOM Message		Corresponding Modbus Register				
Index	Number		Hi Byte Contains	Lo Byte Contains	Register	High Byte	Low Byte
0	1	==>	Byte 0	Status	424833	Number of remaining messages = 42	*Status
1	1	==>	Byte 2	Byte 1	424834	FW version	FW revision
2	2	==>	Byte 0	Status	424835	1:TDES enabled	*Status
3	2	==>	Byte 2	Byte 1	424836	1:TDEN enabled	1:TDNE enabled
4	3	==>	Byte 0	Status	424837	1:TDEC enabled	*Status
5	3	==>	Byte 2	Byte 1	424838	1:Source 2 over frequency monitoring enabled	1:Source 2 under frequency monitoring enabled
6	4	==>	Byte 0	Status	424839	1:Source 2 over voltage monitoring enabled	*Status
7	4	==>	Byte 2	Byte 1	424840	Disable Transfer Delay Bypass (0, 1, 2, 3)	1:Source 2 under voltage monitoring enabled
8	5	==>	Byte 0	Status	424841	1:User selectable preferred source	*Status
9	5	==>	Byte 2	Byte 1	424842	1:Source 1 under frequency monitoring enabled	1:Plant Exerciser enabled
10	6	==>	Byte 0	Status	424843	1:Source 1 over frequency monitoring enabled	*Status
11	6	==>	Byte 2	Byte 1	424844	Operation (0:auto, 1:user selectable)	1:Source 1 over voltage monitoring enabled
12	7	==>	Byte 0	Status	424845	1:TDN enabled	*Status
13	7	==>	Byte 2	Byte 1	424846	1:Pre-transfer signal on subnet enabled	1:TDN Load Sense Delay enabled
14	8	==>	Byte 0	Status	424847	1:Remote sequencing on subnet enabled	*Status

15	8	==>	Byte 2	Byte 1	424848	1:Overcurrent protection enabled	1:Service entrance enabled
16	9	==>	Byte 0	Status	424849	Type of switch: 0:SPB, 1:motor driven	*Status
17	9	==>	Byte 2	Byte 1	424850	1:Load shed from source 2 available	1:PT ratio is user selectable
18	10	==>	Byte 0	Status	424851	1:Source 1 area protection enabled	*Status
19	10	==>	Byte 2	Byte 1	424852	Reserved	0:open transfer only available, 1: also inphase, 2: also inphase or closed
20	11	==>	Byte 0	Status	424853	TDES timer - low byte	*Status
21	11	==>	Byte 2	Byte 1	424854	TDNE timer - low byte	TDES timer - high byte
22	12	==>	Byte 0	Status	424855	TDNE timer - high byte	*Status
23	12	==>	Byte 2	Byte 1	424856	TDEN timer - high byte	TDEN timer - low byte
24	13	==>	Byte 0	Status	424857	TDEC timer - low byte	*Status
25	13	==>	Byte 2	Byte 1	424858	Nominal frequency - low byte	TDEC timer - high byte
26	14	==>	Byte 0	Status	424859	Nominal frequency - high byte	*Status
27	14	==>	Byte 2	Byte 1	424860	Nominal voltage - high byte	Nominal voltage - low byte
28	15	==>	Byte 0	Status	424861	Source 1 undervoltage drop out (volts - low byte)	*Status
29	15	==>	Byte 2	Byte 1	424862	Source 2 undervoltage drop out (volts - low byte)	Source 1 undervoltage drop out (volts - high byte)
30	16	==>	Byte 0	Status	424863	Source 2 undervoltage drop out (volts - high byte)	*Status
31	16	==>	Byte 2	Byte 1	424864	Source 1 undervoltage pick up (volts - high byte)	Source 1 undervoltage pick up (volts - low byte)
32	17	==>	Byte 0	Status	424865	Source 2 undervoltage pick up (volts - low byte)	*Status
33	17	==>	Byte 2	Byte 1	424866	Source 1 overvoltage drop out (volts - low byte)	Source 2 undervoltage pick up level (volts - high byte)
34	18	==>	Byte 0	Status	424867	Source 1 overvoltage drop out (volts - high byte)	*Status
35	18	==>	Byte 2	Byte 1	424868	Source 2 overvoltage drop out (volts - high byte)	Source 2 overvoltage drop out (volts - low byte)
36	19	==>	Byte 0	Status	424869	Source 1 overvoltage pick up (volts - low byte)	*Status
37	19	==>	Byte 2	Byte 1	424870	Source 2 overvoltage pick up (volts - low byte)	Source 1 overvoltage pick up (volts - high byte)
38	20	==>	Byte 0	Status	424871	Source 2 overvoltage pick up (volts - high byte)	*Status
39	20	==>	Byte 2	Byte 1	424872	Source 1 underfrequency drop out (Hz x 10) - high byte	Source 1 underfrequency drop out (Hz x 10) - low byte
40	21	==>	Byte 0	Status	424873	Source 2 underfrequency drop out (Hz x 10) - low byte	*Status
41	21	==>	Byte 2	Byte 1	424874	Source 1 underfrquency pick up (Hz x 10) - low byte	Source 2 underfrquency drop out (Hz x 10) - high byte
42	22	==>	Byte 0	Status	424875	Source 1 underfrequency pick up (Hz x 10) - high byte	*Status
43	22	==>	Byte 2	Byte 1	424876	Source 2 underfrequency pick up (Hz x 10) - high byte	Source 2 underfrequency pick up (Hz x 10) - low byte
44	23	==>	Byte 0	Status	424877	Source 1 overfrequency drop out (Hz x 10) - low byte	*Status
45	23	==>	Byte 2	Byte 1	424878	Source 2 overfrequency drop out (Hz x 10) - low byte	Source 1 overfrequency drop out (Hz x 10) - high byte
46	24	==>	Byte 0	Status	424879	Source 2 overfrequency drop out (Hz x 10) - high byte	*Status
47	24	==>	Byte 2	Byte 1	424880	Source 1 overfrequency pick up (Hz x 10) - high byte	Source 1 overfrequency pick up (Hz x 10) - low byte
48	25	==>	Byte 0	Status	424881	Source 2 overfrequency pick up (Hz x 10) - low byte	*Status
49	25	==>	Byte 2	Byte 1	424882	1: TDN with Load Sensing enabled - low byte	Source 2 overfrequency pick up (Hz x 10) - high byte
50	26	==>	Byte 0	Status	424883	TDN with Loading Sensing enabled - high byte	*Status
51	26	==>	Byte 2	Byte 1	424884	TDN timer - high byte	TDN timer - low byte
52	27	==>	Byte 0	Status	424885	Load voltage decay threshold - low byte	*Status
53	27	==>	Byte 2	Byte 1	424886	0,1,2: Preferred Source selection - low byte	Load voltage decay threshold - high byte
54	28	==>	Byte 0	Status	424887	Preferred Source selection - high byte	*Status

55	28	==>	Byte 2	Byte 1	424888	Plant exerciser enabled - high byte	1: Plant exerciser enabled - low byte
56	29	==>	Byte 0	Status	424889	1: Plant exerciser load transfer enabled - low byte	*Status
57	29	==>	Byte 2	Byte 1	424890	Plant exerciser day of week - low byte	Plant exerciser load transfer enabled - high byte
58	30	==>	Byte 0	Status	424891	Plant exerciser day of week - high byte	*Status
59	30	==>	Byte 2	Byte 1	424892	Plant exerciser hour of day - high byte	Plant exerciser hour of day - low byte
60	31	==>	Byte 0	Status	424893	Plant exerciser minute - low byte	*Status
61	31	==>	Byte 2	Byte 1	424894	Manual re-transfer mode - low byte (0: auto, 1: pushbutton return)	Plant exerciser minute - high byte
62	32	==>	Byte 0	Status	424895	Manual re-transfer mode - high byte	*Status
63	32	==>	Byte 2	Byte 1	424896	Commit to transfer in TDNE - high byte	Commit to transfer in TDNE - low byte (1: commit)
64	33	==>	Byte 0	Status	424897	Test mode engine start only - low byte (0: no load transfer)	*Status
65	33	==>	Byte 2	Byte 1	424898	Engine run test time (minutes) - low byte	Test mode engine start only - high byte
66	34	==>	Byte 0	Status	424899	Engine run test time (minutes - high byte	*Status
67	34	==>	Byte 2	Byte 1	424900	Subnetwork pre-transfer time (seconds) - low byte	Subnetwork pre-transfer time (seconds) - high byte
68	35	==>	Byte 0	Status	424901	Number of generators - low byte (0-2)	*Status
69	35	==>	Byte 2	Byte 1	424902	Three phase or single phase - low byte	Number of generators - high byte
70	36	==>	Byte 0	Status	424903	Three phase or single phase - high byte	*Status
71	36	==>	Byte 2	Byte 1	424904	Subnetwork sequencing timer (seconds) - high byte	Subnetwork sequencing timer (seconds) - low byte
72	37	==>	Byte 0	Status	424905	PT ratio - low byte	*Status
73	37	==>	Byte 2	Byte 1	424906	1: Closed transition transfer enabled - low byte	PT ratio - high byte
74	38	==>	Byte 0	Status	424907	Closed transition transfer enabled - high byte	*Status
75	38	==>	Byte 2	Byte 1	424908	Closed transition allowed phase angle difference - low byte	Closed transition phase allowed phase angle difference - high byte
76	39	==>	Byte 0	Status	424909	Closed transition allowed frequency difference - low byte	*Status
77	39	==>	Byte 2	Byte 1	424910	Closed transition allowed voltage difference - low byte	Closed transition allowed frequency difference (high byte)
78	40	==>	Byte 0	Status	424911	Closed transition allowed voltage difference - high byte	*Status
79	40	==>	Byte 2	Byte 1	424912	In phase transition enabled - high byte	1: In phase transition enabled
80	41	==>	Byte 0	Status	424913	In phase transition allowed phase angle difference - low byte	*Status
81	41	==>	Byte 2	Byte 1	424914	In phase transition allowed frequency difference - low byte	
82	42	==>	Byte 0	Status	424915	In phase transition allowed frequency difference - high byte	*Status
83	42	==>	Byte 2	Byte 1	424916	Maximum allowed time to attempt synchronization - high byte	Maximum allowed time to attempt synchronization - low byte
84	43	==>	Byte 0	Status	424917	Checksum of previous 42 messages - low byte	*Status
85	43	==>	Byte 2	Byte 1	424918	1's complement of high byte of value contained in 424917	Checksum of previous 42 messages - high byte

IQ Transfer: Function (3 F 9) Download (Write) Setpoints Buffer

Modbus MINT Cross-Reference

Operation: The Modbus master transmits 43 separate messages, each message containing 2 Modbus registers

Send this Modbus register pair first. This transmits the 3 F 9 command (INST = 3, COMM = F, SCOMM = 9)

Index	**INCOM Message** Number		**Corresponding Modbus Register** Hi Byte Contains	Lo Byte Contains	Register	High Byte	Low Byte
0	1	==>	Byte 0	Control	424577	1111 0011 = F3h = 243 decimal	1000 0001 = 81h = 129 decimal
1	1	==>	Byte 2	Byte 1	424578	1001 0000 = 90h = 144 decimal	0000 0001 = 01h = 1 decimal

Byte 0

bits 0 - 3 : INST field (3)

bits 4 - 7 : COMM field (F)

Control

bits 0 - 5 : number of expected response messages from device

(in this example we expect 1 message [ACK] as reply)

bit 6 : 0 (reserved)

bit 7 : 0 - data message

1 - control messages

Byte 2

bits 0 - 3 : bits 8 - 11 of address of device (*0000* 0000 0001b)

bits 4 - 7 : SCOMM field (9)

Byte 1

bits 0 - 7 : bits 0 - 7 of address of device (0000 *0000 0001*b)

Next send these messages, the data exactly follows the received setpoint data returned with the 3 C 9 command. Edit fields as required and transmit to the mMINT in this sequence.
Subequent messages are transmitted as soon as an acknowledgment is received from the previous message.

	INCOM Message Number		**Corresponding Modbus Register** Hi Byte Contains	Lo Byte Contains	Register	High Byte	Low Byte
0	1	==>	Byte 0	Status	424577	Number of remaining messages = 42	0000 0001 = 01h = 1 decimal
1	1	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
2	2	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
3	2	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
4	3	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
5	3	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
6	4	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
7	4	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
8	5	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal

9	5	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
10	6	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
11	6	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
12	7	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
13	7	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
14	8	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
15	8	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
16	9	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
17	9	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
18	10	==>	Byte 0	Status	424577	Don't Care	0000 0001 = 01h = 1 decimal
19	10	==>	Byte 2	Byte 1	424578	Don't Care	Don't Care
20	11	==>	Byte 0	Status	424577	TDES timer - low byte	0000 0001 = 01h = 1 decimal
21	11	==>	Byte 2	Byte 1	424578	TDNE timer - low byte	TDES timer - high byte
22	12	==>	Byte 0	Status	424577	TDNE timer - high byte	0000 0001 = 01h = 1 decimal
23	12	==>	Byte 2	Byte 1	424578	TDEN timer - high byte	TDEN timer - low byte
24	13	==>	Byte 0	Status	424577	TDEC timer - low byte	0000 0001 = 01h = 1 decimal
25	13	==>	Byte 2	Byte 1	424578	Nominal frequency - low byte	TDEC timer - high byte
26	14	==>	Byte 0	Status	424577	Nominal frequency - high byte	0000 0001 = 01h = 1 decimal
27	14	==>	Byte 2	Byte 1	424578	Nominal voltage - high byte	Nominal voltage - low byte
28	15	==>	Byte 0	Status	424577	Source 1 undervoltage drop out (volts - low byte)	0000 0001 = 01h = 1 decimal
29	15	==>	Byte 2	Byte 1	424578	Source 2 undervoltage drop out (volts - low byte)	Source 1 undervoltage drop out (volts - high byte)
30	16	==>	Byte 0	Status	424577	Source 2 undervoltage drop out (volts - high byte)	0000 0001 = 01h = 1 decimal
31	16	==>	Byte 2	Byte 1	424578	Source 1 undervoltage pick up (volts - high byte)	Source 1 undervoltage pick up (volts - low byte)
32	17	==>	Byte 0	Status	424577	Source 2 undervoltage pick up (volts - low byte)	0000 0001 = 01h = 1 decimal
33	17	==>	Byte 2	Byte 1	424578	Source 1 overvoltage drop out (volts - low byte)	Source 2 undervoltage pick up level (volts - high byte)
34	18	==>	Byte 0	Status	424577	Source 1 overvoltage drop out (volts - high byte)	0000 0001 = 01h = 1 decimal
35	18	==>	Byte 2	Byte 1	424578	Source 2 overvoltage drop out (volts - high byte)	Source 2 overvoltage drop out (volts - low byte)

36	19	==>	Byte 0	Status	424577	Source 1 overvoltage pick up (volts - low byte)	0000 0001 = 01h = 1 decimal
37	19	==>	Byte 2	Byte 1	424578	Source 2 overvoltage pick up (volts - low byte)	Source 1 overvoltage pick up (volts - high byte)
38	20	==>	Byte 0	Status	424577	Source 2 overvoltage pick up (volts - high byte)	0000 0001 = 01h = 1 decimal
39	20	==>	Byte 2	Byte 1	424578	Source 1 underfrequency drop out (Hz x 10) - high byte	Source 1 underfrequency drop out (Hz x 10) - low byte
40	21	==>	Byte 0	Status	424577	Source 2 underfrequency drop out (Hz x 10) - low byte	0000 0001 = 01h = 1 decimal
41	21	==>	Byte 2	Byte 1	424578	Source 1 underfrquency pick up (Hz x 10) - low byte	Source 2 underfrquency drop out (Hz x 10) - high byte
42	22	==>	Byte 0	Status	424577	Source 1 underfrequency pick up (Hz x 10) - high byte	0000 0001 = 01h = 1 decimal
43	22	==>	Byte 2	Byte 1	424578	Source 2 underfrequency pick up (Hz x 10) - high byte	Source 2 underfrequency pick up (Hz x 10) - low byte
44	23	==>	Byte 0	Status	424577	Source 1 overfrequency drop out (Hz x 10) - low byte	0000 0001 = 01h = 1 decimal
45	23	==>	Byte 2	Byte 1	424578	Source 2 overfrequency drop out (Hz x 10) - low byte	Source 1 overfrequency drop out (Hz x 10) - high byte
46	24	==>	Byte 0	Status	424577	Source 2 overfrequency drop out (Hz x 10) - high byte	0000 0001 = 01h = 1 decimal
47	24	==>	Byte 2	Byte 1	424578	Source 1 overfrequency pick up (Hz x 10) - high byte	Source 1 overfrequency pick up (Hz x 10) - low byte
48	25	==>	Byte 0	Status	424577	Source 2 overfrequency pick up (Hz x 10) - low byte	0000 0001 = 01h = 1 decimal
49	25	==>	Byte 2	Byte 1	424578	1: TDN with Load Sensing enabled - low byte	Source 2 overfrequency pick up (Hz x 10) - high byte
50	26	==>	Byte 0	Status	424577	TDN with Loading Sensing enabled - high byte	0000 0001 = 01h = 1 decimal
51	26	==>	Byte 2	Byte 1	424578	TDN timer - high byte	TDN timer - low byte
52	27	==>	Byte 0	Status	424577	Load voltage decay threshold - low byte	0000 0001 = 01h = 1 decimal
53	27	==>	Byte 2	Byte 1	424578	0,1,2: Preferred Source selection - low byte **	Load voltage decay threshold - high byte
54	28	==>	Byte 0	Status	424577	Preferred Source selection - high byte	0000 0001 = 01h = 1 decimal
55	28	==>	Byte 2	Byte 1	424578	Plant exerciser enabled - high byte	1: Plant exerciser enabled - low byte
56	29	==>	Byte 0	Status	424577	1: Plant exerciser load transfer enabled - low byte	0000 0001 = 01h = 1 decimal
57	29	==>	Byte 2	Byte 1	424578	Plant exerciser day of week - low byte	Plant exerciser load transfer enabled - high byte
58	30	==>	Byte 0	Status	424577	Plant exerciser day of week - high byte	0000 0001 = 01h = 1 decimal
59	30	==>	Byte 2	Byte 1	424578	Plant exerciser hour of day - high byte	Plant exerciser hour of day - low byte
60	31	==>	Byte 0	Status	424577	Plant exerciser minute - low byte	0000 0001 = 01h = 1 decimal
61	31	==>	Byte 2	Byte 1	424578	Manual re-transfer mode - low byte (0: auto, 1: pushbutton return)	Plant exerciser minute - high byte

62	32	==>	Byte 0	Status	424577	Manual re-transfer mode - high byte	0000 0001 = 01h = 1 decimal
63	32	==>	Byte 2	Byte 1	424578	Commit to transfer in TDNE - high byte	Commit to transfer in TDNE - low byte (1: commit)
64	33	==>	Byte 0	Status	424577	Test mode engine start only - low byte (0: no load transfer)	0000 0001 = 01h = 1 decimal
65	33	==>	Byte 2	Byte 1	424578	Engine run test time (minutes) - low byte	Test mode engine start only - high byte
66	34	==>	Byte 0	Status	424577	Engine run test time (minutes - high byte	0000 0001 = 01h = 1 decimal
67	34	==>	Byte 2	Byte 1	424578	Subnetwork pre-transfer time (seconds) - low byte	Subnetwork pre-transfer time (seconds) - high byte
68	35	==>	Byte 0	Status	424577	Number of generators - low byte (0-2)	0000 0001 = 01h = 1 decimal
69	35	==>	Byte 2	Byte 1	424578	Three phase or single phase - low byte	Number of generators - high byte
70	36	==>	Byte 0	Status	424577	Three phase or single phase - high byte	0000 0001 = 01h = 1 decimal
71	36	==>	Byte 2	Byte 1	424578	Subnetwork sequencing timer (seconds) - high byte	Subnetwork sequencing timer (seconds) - low byte
72	37	==>	Byte 0	Status	424577	PT ratio - low byte	0000 0001 = 01h = 1 decimal
73	37	==>	Byte 2	Byte 1	424578	1: Closed transition transfer enabled - low byte	PT ratio - high byte
74	38	==>	Byte 0	Status	424577	Closed transition transfer enabled - high byte	0000 0001 = 01h = 1 decimal
75	38	==>	Byte 2	Byte 1	424578	Closed transition allowed phase angle difference - low byte	Closed transition phase allowed phase angle difference - high byte
76	39	==>	Byte 0	Status	424577	Closed transition allowed frequency difference - low byte	0000 0001 = 01h = 1 decimal
77	39	==>	Byte 2	Byte 1	424578	Closed transition allowed voltage difference - low byte	Closed transition allowed frequency difference (high byte)
78	40	==>	Byte 0	Status	424577	Closed transition allowed voltage difference - high byte	0000 0001 = 01h = 1 decimal
79	40	==>	Byte 2	Byte 1	424578	In phase transition enabled - high byte	1: In phase transition enabled
80	41	==>	Byte 0	Status	424577	In phase transition allowed phase angle difference - low byte	0000 0001 = 01h = 1 decimal
81	41	==>	Byte 2	Byte 1	424578	In phase transition allowed frequency difference - low byte	
82	42	==>	Byte 0	Status	424577	In phase transition allowed frequency difference - high byte	0000 0001 = 01h = 1 decimal
83	42	==>	Byte 2	Byte 1	424578	Maximum allowed time to attempt synchronization - high byte	Maximum allowed time to attempt synchronization - low byte
84	43	==>	Byte 0	Status	424577	Checksum of previous 42 messages - low byte	0000 0001 = 01h = 1 decimal
85	43	==>	Byte 2	Byte 1	424578	1's complement of high byte of value contained in 424917	Checksum of previous 42 messages - high byte

** Preferred Source Selection Values:

0: No preferred source. Load remains connected to present source until that source is unavailable

1: Source 1 is preferred. Operation on Source 2 is temporary and only occurs while Source 1 is unavailable

2: Source 2 is preferred. Operation on Source 1 is temporary and only occurs while Source 2 is unavailable

Note: switching preferred source changes the meaning of "Normal" (preferred) and "Emergency" (not preferred) sources.
Therefore timers associated with normal and emergency source will associate with preferred and not preferred sources, respectively.

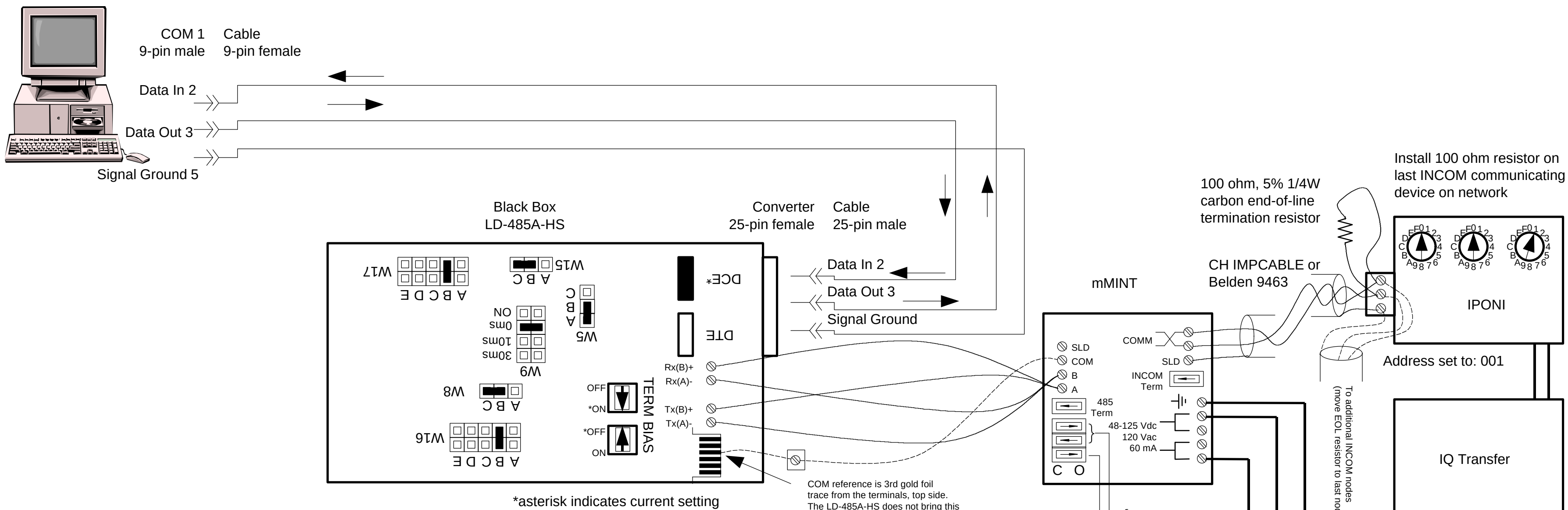
Example:

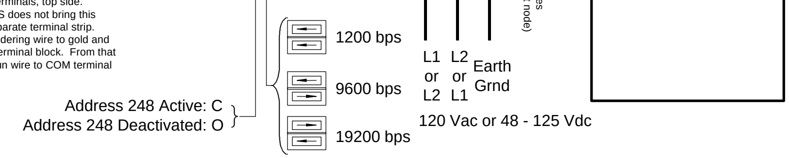
Source 1 is preferred

TDNE = 5 seconds

TDEN = 10 minutes

These timer settings imply a 5 second delay after generator availalable (and preferred source not availlable) before switching to generator.
If a command is received that switches the preferred source to the generator, the IQ Transfer will note correctly that the load is now connected to the non-preferred source and begin to time to return to the preferred source using the TDEN timer. This will require 10 minutes.
This time delay may be considered excessive. The solution is to either transmit the dispatching command early by the same number of minutes of the TDEN timer or reduce the TDEN timer to a smaller value.





W5 Hardware Handshake
A-B RTS/CTS (*)
B-C RTS/CTS/CD

W8 Physical Connection
A-B 4-wire
B-C 2-wire (*)

W9 RTS/CTS Delay
A 30 ms
B 10 ms
C 0 ms (*)
D CTS always on
(lab testing indicates that any setting will work)

W15 Turnaround Delay
A-B RTS/CD enabled transmitter
B-C Data enabled transmitter (*)

W16 Turnaround Delay
A 0 ms
B 0.1 ms (*)
C 1 ms
D 5 ms
E 35 ms
(lab testing shows that 0 - 35 ms will work unless the mMINT returns an exception. In that case, only 0 - 5 ms will work).

W17 Driver Enabled Hold
A 70 ms
B 7 ms (*)
C 2 ms
D 0.7 ms
E 0.15 ms
(lab testing shows that any of these settings will work, but the jumper must be installed on at least one setting)

<u>mMINT set to (as shown):</u>
Diagnostic address 248 activated
RS-485 set to 19200 bps
RS-485 end-of-line termination set on (closed)
INCOM end-of-line termination set on (closed)

1 REVISION	THE INFORMATION ON THIS DOCUMENT IS CREATED BY CUTLER-HAMMER. IT IS DISCLOSED IN CONFIDENCE AND IS ONLY TO BE USED FOR THE PURPOSE IN WHICH IT IS SUPPLIED	DFTR / DATE	Cutler-Hammer	BEAVER, PA
		APPD DGL / DATE 11-MAR-02	TITLE	mMINT Interface Diagram
		APPD / DATE	TYPE	
	FEDERAL ID NO 341756464	REVISION 2 / DWG SIZE A	G.O. - / DWG -	Sheet 1 OF 1