Genetix GCM Configuration and Troubleshooting for Modbus

Both serial ports in a Genetix GCM can be configured to support Modbus ASCII or Modbus RTU. This allows for establishing direct communications to almost any industrial network, either directly or using protocol converters. We currently support interfacing to Profibus DP, EtherNet/IP, Modbus/TCP, Modbus Plus, DeviceNet or ControlNet using this method.

The network host can take control over and/or monitor the GCM, using Common Interface Table (CIT) data exchange. For details, see http://www2.merrick-inc.com/mct/CIT.pdf

System Requirements

All versions of the Merrick Genetix GCM controller support Mobus ASCII and Modbus RTU communications protocols. They will act as slaves, supporting Modbus functions 3 (Read Holding Registers) and 16 (Preset Multiple Registers). Function 8, sub function 0 (Return Query Data) is also supported.

The physical interface is 4-wire RS-485 or 3-wire RS232. The master is assumed to maintain a cyclic conversation with the controllers, which exposes a CIT, making it possible to monitor and/or supervise the GCM completely.

Different data types are used for control/status bits, integer numbers and floating-point numbers. Control/Status bits and Integer numbers are organized in 16 bit words. Parameters are organized as IEEE 32 bit floating point numbers, located in two consecutive 16 bit words. For PLC's that don't support floating point numbers, it is possible to split parameters into two 16 bit integers, one containing the integer part and one containing the fractional part multiplied by 10,000 (four implied decimal places).

Configuring the GCM

Configuring the GCM includes setting up communications parameters, register tags, warnings, faults and external inputs and outputs. The register tags are set to make any internal GCM parameter appear in Tag 1 R to Tag 5 R values. Standard feeder parameters are always available. Warnings and faults are user preference qualified, associated with any logical I/O point in the GCM. This configuration is done regardless if communications is used or not. Any logical inputs you want to control from the PLC must be mapped to a network input. In the same way, logical outputs must be mapped to network outputs for monitoring purposes.

It is recommended that you use the built in Terminal Monitoring and Configuration Interface (TMCI) for setting up, troubleshooting and monitoring the Modbus interface. You need a PC with a terminal emulator supporting the VT52 terminal protocol, i.e. Hyperterminal, a USB port and a type A USB cable. Merrick’s service department can help you set this up.The following menu references and screen shots were taken using the GCM 101.1.7.2 Belt Feeder Controller application. Operation and Maintenance Manuals as well as register specifications are available at the Merrick Web Site:

http://www2.merrick-inc.com/mct/GENApps/GENApps.htm

Setting the GCM communications parameters

Set up the serial port in TMCI page X (Type PX<Enter>). Select which of the two serial ports (port 1 or 2) you will use. Port 1 is normally used for a user interface. Rotate between ports using X<Enter>.

```
BLT>
                PX  X: Port 2 (UART1) settings

A: NewProt     2|LastProt    2|B: BdIdx    5|C: LnPar 0x03|
    MB RTU      | MB RTU      | 19200       |             |

En:WProt  0x0BFF|WOrder 0x0000|IntFra 0x0000|
Fn:Tag 1       0|Tag 2       0|Tag 3       0|Tag 4       0|Tag 5       0

   Protocol     |1:  Unit Addr|2:   Map Size|3:   Max Time|4:         LRC
Gn:All Modbus   |            1|            1|        10.00|
Hn:DF-1         |            0|            1|        10.00|           0
In:CITMaster    |           49|             |         2.00|
Jn:UISServer    |            1|             |        20.00|
Kn:Scalenet     |           49|            1|        10.00|

   Scalenet Spec|   Start Char|     End Char| No Powup Hnd|
                |L:       0x0A|M:       0x0D|N:          1|
   Mon Period ms|O:        500|             |             |

R: Restart port
```

PX, TMCI Port Settings Screen

Parameter	Comment
A : NewProt	Protocol Selector. 1 for Modbus ASCII, 2 for Modbus RTU
LastProt	Last and currently used protocol. Unchanged until you restart the port.
B: BdIdx	Index of the baud rate used: 0 300 1 1200 2 2400 3 4800 4 9600 5 19200 6 38400 7 57600 8 115200 9 230400 Set to the highest baudrate supported by the Modbus master and all slaves. Use 6, 34800 baud for Anybus Communicators.

Parameter	Comment
C: LnPar	UART Line parameter settings, hexadecimal. Bit encoded: Bit 0, 1 Word Length – 00: 5 bits, 01: 6 bits, 10: 7 bits, 11: 8 bits. Bit 2 Stop bits. 0: 1 stop bit, 1: 2 stop bits Bit 3 Parity Enable Bit 4, 5 Parity Select – 00: Odd, 01: Even, 10: Forced 1, 11: Forced 0 Bit 6 Enable Break transmission. Always 0! Bit 7 DLAB Access. Always 0! Typical settings are 0x03 for N, 8, 1 (most commonly used) and 0x1B for E,8,1 (Modbus RTU default)
E1:WProt	Write Protection bits, hexadecimal. Normally 0xBFF, meaning all tags except the Primary Setpoint are write protected. See CIT documentation.
E2:Worder	Word Order bits, Hexadecimal. Normally 0x0000, meaning all tags have normal word order. This has to be found by trial and error. The Siemens S7 PLC, for example, has the word order for a floating point value reversed (as compared with a Modicon PLC), so you must enter 0x0FFF. See CIT documentation.
E3:IntFra	Integer/Fraction bits, hexadecimal. Normally 0x0000, meaning all tags are transferred as Floating Point numbers. See CIT documentation.
Fn:Tag n, n=1..5	Tagged register number. Up to five internal GCM Register Numbers you want to monitor or set. The corresponding register value will appear in the "Tag 1 Rd V" position in the CIT. Register numbers are available in the GCM Firmware Overview.
G1: Unit Addr	Modbus “Node Number” Must be different for all slaves on the multidrop serial line.
G2: Map Size	There are two different CIT layouts. This document covers CIT type 1 only. Always set to 1!
G3: Max Time	Max allowed time between incoming, valid Modbus telegrams addressing the Unit Address of this GCM. If this time is exhausted, the logical output Port 1 Timeout or Port 2 Timeout goes on, depending on which port is used for Modbus communications.

If you extend the size of the write table beyond word 59, you will overwrite “Write Protect”, “Word Order”, “Int/Frac FP” and "Tag Reg 1-5" parameters from the PLC. This allows for multiplexing of the parameters in the "Tag n Rd V" positions.

Configuring warnings and faults

Warnings and Faults are qualifiers to logical inputs and outputs, normally set by the user. See “Digital I/O mapping” in the Operations and Maintenance manual. Warnings are considered to require attention. Faults are considered to be fatal for the feeder operation, and the controller will attempt to stop the feeder. In this example the warnings and faults are set up according to the following table:

I/O point	Qualifier	Comment
LoadCell 1 Ovl	Fault	Invalid Load Cell Signal
LoadCell 1 Undl	Fault	Invalid Load Cell Signal
Blt Drive Fail	Fault	Time delayed indication that the belt drive has failed or is not ready.
High Belt Load	Warning	Too much material on the belt
Belt Drv Ovld	Warning	Signal from the belt motor drive, connected to an MC³

The warning and faults are normally set up using the user interface; **Advanced Setup, Digital I/O, Warings/Faults**.

Warning bits (CIT word 18) and Fault bits (CIT word 19) go on whenever the corresponding logical inputs or outputs go on and stay on until they are deliberately cleared. Faults and Warnings can be cleared using the user interface or TMCI page K. Warnings can also be cleared using CIT word 44, bit 8. See CIT documentation. TCMI page K looks like this

```
BLT>
              PK  Belt_1_7_2 Faults and Warnings

Warnings: 0x00000002, W to clear.   Faults 0x00000000, F to clear
012 0 0 0 High Belt Load      |006 0 0 0 LoadCell 1 Ovl
158 1 1 0 Belt Drv Ovld       |007 0 0 0 LoadCell 1 Undl
                              |050 0 0 0 Belt Drive Fail
```

`Warnings: 0x00000002` – The hex value of Warnings, CIT word 18.

`Faults: 0x00000000` – The hex value of Faults, CIT word 19.

There are five columns for each Fault and Warning:

Col 1: Internal number of the logical input or output

Col 2: State of the logical input or output

Col 3: State of the corresponding bit in Warnings or Faults in the CIT

Col 4: Not used in this mapping

Col 5: Name of the logical input or output.

It is important to note that the bits in the Warnings and Faults words reflect the state of "1" in the third column, not the second. In this example, both are on for the logical input **Belt Drv Ovld**. Bit 1 of the Warnings word is on. If the **Belt Drv Ovld** input is turned off, then the "1" in the second column goes away, but stays in the third until the warning is cleared, either from the user interface or by the "Clear Warnings Command" bit [8] in the Control [44] register.

Note that the bit order in the CIT words for Warnings and Faults are not configurable. It is derived from the order of logical outputs and logical inputs in the Digital Outputs and Digital Inputs screens. If you add or remove a warning or fault qualifier to a logical output or input, the bit position changes.

Configuring Network Inputs and Outputs

Logical inputs and outputs can be mapped in three ways:

1. To a physical input or output. In this example the logical input **Belt Drv Ovld** is mapped to **GCM Input 2**, which, in turn, is connected to the Overload output of the belt motor VFD. The physical output **GCM Output 1** is mapped to the logical output **Fdr Drv Enable**. The output is then connected to the Start input of the belt motor VFD.
2. To a network input or output. In this example, the logical input **Run Permission** is mapped to **Network Input 1**. This allows the PLC to control (start and stop) the feeder through the External Inputs word 45, bit 0.
3. Unused Logical Inputs are typically connected to the Physical Input **Always On** or **Always Off**.

The PLC controls network inputs to the GCM in word 45 in the CIT. They appear in the GCM as **Netw Input 1**.. **Netw Input 16**. You then map them to Logical Inputs in the GCM. Note that the

Network Inputs are numbered 1 - 16. Bits in the corresponding PLC word are typically numbered 0 - 15. A Modicon PLC will use bit 16 – 1.

It is possible to have a physical input wrapped around to an network output (for monitoring purposes) by first map the input to an 'Available I/O' point, and then map the same Available I/O point to a network output.

In this example, we want logical I/O points that can cause a Warning or Fault visible in the Network Outputs register (CIT Word 17), along with the logical outputs Ready, Running and Good Feedrate. We only control one input, Run Permission.

This is how the digital inputs were mapped in the GCM for this example:

Logical	Physical	Warning/Fault	Word:Bit in the CIT
Run Permission	Network Input 1		45:0 to GCM
Feeder Block	GCM Input 1		41:8 from GCM
Belt Drv Ovld	GCM Input 2	Warning	41:9 from GCM
Belt Drive Fail	GCM Input 3	Fault	41:10 from GCM

Digital output mapping:

Physical	Logical	Warning/Fault	Word:Bit in the CIT
GCM Output 1	Fdr Drv Enable		42:8 from GCM
External Output 1	Fault		17:0 from GCM
External Output 2	Warning		17:1 from GCM
External Output 3	Ready		17:2 from GCM
External Output 4	Hi Belt Load	Warning	17:3 from GCM
External Output 5	LoadCell 1 Ovl	Fault	17:4 from GCM
External Output 6	LoadCell 1 Undl	Fault	17:5 from GCM

Setting Up the Setpoint Source

The GCM Setpoint Source must be set to Network (Serial): Advanced Setup, Feeder Control, Source. The setpoint is taken from CIT word 46 and 47.

Connect the Host or Interface to the GCM and Check Communications

Use a cable designed for RS-422. There should be two pairs, individually shielded, with a characteristic impedance of around 150Ω. Belden 9368 or equivalent is a good alternative. Add 121Ω terminating resistors at both ends of the RS485 line. Connect one between Rx+ and Rx- and one between terminals Tx+ and TX- at each end of the cable for a total of four resistors. The pin numbering on the GCM is left to right as you look at the serial connector. See the picture on the next page. The GCM receives data on terminals 6 and 7 (Rx+ and Rx-), and transmits, after being correctly addressed, on terminals 8 and 9 (Tx+ and Tx-).

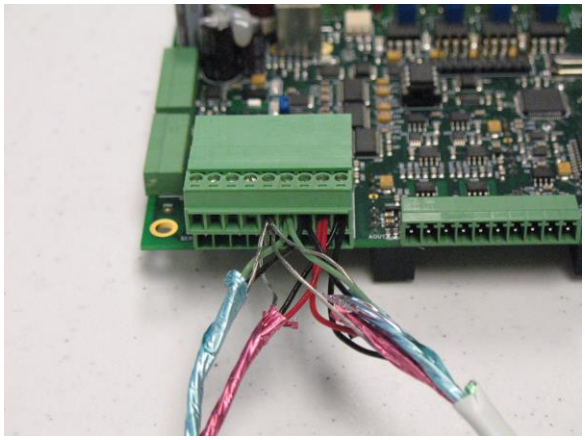
Pair	Part	GCM Terminal	RS-422 Name
		5	Shield
1	1	6	Rx+
1	2	7	Rx-
2	1	8	Tx+
2	2	9	Tx-

The RS485 interface allows for up to 50V common mode swing with respect to the 24V DC negative terminal on the GCM.

You can also use the RS-232 interface if the GCM is the only slave on the line, effectively a point-to-point connection. The cable must be short, less than 50 feet at 19200 baud. The GCM receives data on terminal 2 (RxD) with respect to GND, and transmits, after being addressed, on terminal 3 (TxD) with respect to GND. GND is terminal 4.

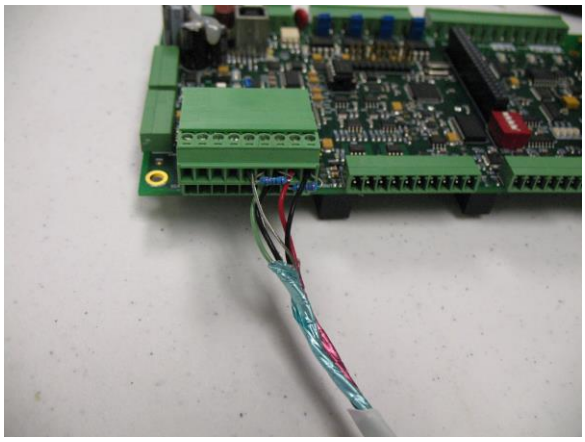
The RS232 interface also allows for up to 50V common mode swing with respect to the 24V DC negative terminal on the GCM.

Both ports have LED indicators for receive (Green) and transmit (Red) data. Port 2 LEDs are to the right on the GCM board.



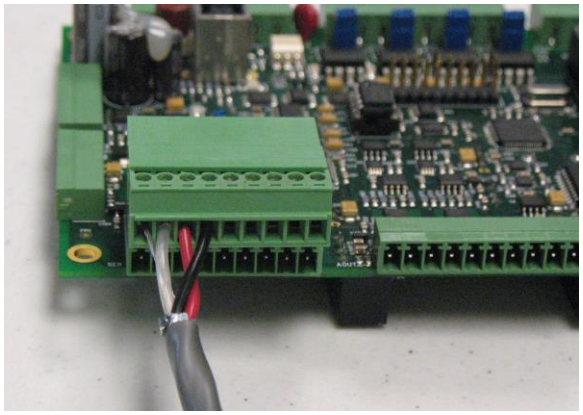
This picture shows a GCM connected to an inner point of a 4-wire RS485 multidrop line. Note that the cable used has two individually shielded pairs. The drain wire from all four pairs (two incoming, two outgoing) are connected to terminal 5, the RS485 cable ground.

The upper row of terminals is Port 2 (UART1) and the lower row is Port 1 (UART0).



This picture shows a GCM connected to an end point of a 4-wire RS485 multidrop line. Note the terminating resistors.

The master does not have to be an end point of the multidrop line, assuming you can disengage the terminating resistors in the master.



This is an RS232 connection. The shield is connected to terminal 1, GCM Chassis Ground. For some hosts, it may be better to connect the shield to pin 4, RS232 GND.

If you are using one of the recommended protocol converters from Anybus, Merrick has an adapter board (Part No M22787-1) that connects to the DB9-S connector and offers screw terminals for the cable.

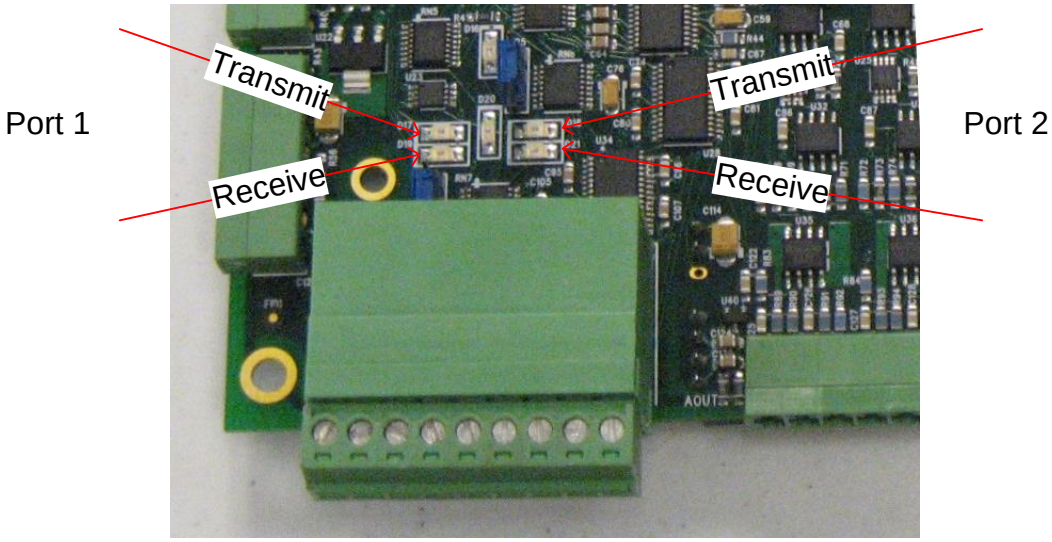
When you have connected and powered up, check that all looks OK in the ModBus Diagnostic screen. See page 8.

Troubleshooting tips

Setting up industrial networks can sometimes be a daunting task. In this example, you may have to deal with several mapping layers and communication protocols. Fortunately, there are excellent troubleshooting tools available.

Look at the LED's on the GCM board.

The transmit (GCM sending) LEDs are red, and the receive LEDs are green. No receive activity indicates cable or host problems. No transmit activity indicates addressing or serial parameter problems



Check Data in the TCMI CIT Diagnostic Screen.

The TCMI CIT Diagnostic Screen (P1) displays data in the CIT as it is received and transmitted over the Modbus port.

```
BLT>
          P1 X: Port 2  (UART1) MB RTU Small Map CIT

16 in[0]  Status/DNI 0080           44 in[0]  Ctl/DNI    0080          [A]
17 in[1]  Ext Outs   0003           45 in[1]  Ext Ins    0001          [B]
18 in[2]  Warnings   0002           46 fl[1]  Feed SP     0.000005e+01 [C]
19 in[3]  Faults     0000           48 fl[2]  Sec SP      0.000000e+00
20 fl[2]  Tag 1 Rd V  0.000000e+00  50 fl[3]  Tag 1 Wr V  0.000000e+00
22 fl[3]  Tag 2 Rd V  0.000000e+00  52 fl[4]  Tag 2 Wr V  0.000000e+00
24 fl[4]  Tag 3 Rd V  0.000000e+00  54 fl[5]  Tag 3 Wr V  0.000000e+00
26 fl[5]  Tag 4 Rd V  0.000000e+00  56 fl[6]  Tag 4 Wr V  0.000000e+00
28 fl[6]  Tag 5 Rd V  0.000000e+00  58 fl[7]  Tag 5 Wr V  0.000000e+00
30 fl[7]  Feedrate    5.000011e+01  60 in[16] WProt Bits 0BFF          [J]
32 fl[8]  Load        9.027183e+00  61 in[17] W Ord Bits 0000          [K]
34 fl[9]  Speed       5.538830e+00  62 in[18] I/F Bits   0000          [L]
36 fl[10] Subtotal    5.171570e+05  63 in[19] Tag 1 Reg     0          [M]
38 fl[11] Total       5.171570e+05  64 in[20] Tag 2 Reg     0          [O]
40 in[24] App/Ver     0165          65 in[21] Tag 3 Reg     0          [Q]
41 in[25] PhIn        5102          66 in[22] Tag 4 Reg     0          [T]
42 in[26] Ph Out L/H 0902/0000      67 in[23] Tag 5 Reg     0          [U]
addr:           44|size:           8|CSErrs:          0|Timee:          0
rxfme:           0|rxlfm:          0|rtgms:        3496|ttgms:        1748
```

Note that the values are only updated on valid Modbus telegrams. If no telegrams have been received, most values are zero. As you can see, the layout follows the CIT exactly. All integer values are presented in hexadecimal format except the Tag register numbers. The ‘e’ format for the floating points can help troubleshooting FP transfers. You are reading from the left column, and writing to the right. If you succeed with the integrity bit, you should see bit 7 in the Sts/DNI and Ctl/DNI toggle.

Check Data in the Communication Diagnostic Screen.

The TCMI Communications Diagnostic Screen (P2) displays communication related data and counters.

```
BLT>
          P2 X: Port 2  (UART1) MB RTU Diagnostics

   NewProt          2|LastProt         2|BaudIdx          4|LinePar       0x1B
   ISRProt          2|Route            0|PrState          0|Port             1
   Ints         27910|Rxchs        63433|Txchs        43997|
   Rxuae            0|Rxlua         0x00|sta           0x61|cha            0xC6
   Rxillmsr         0|Rxilluae         0|Txboverr         0|Rxboverr          0
   Iir           0xC1|Rxtmo         2537|wastmo           0|IllInts           0

   unita            1|                  |Map              1|MaxTime      10.00
   rxsta            0|txsta            2|State            2|isTimeout         0
   addr            16|size            28|cmd              3|subf              0
   cCRC         0xC645|tCRC         0xC645|bcnt             8|lalen             0
   rtgms         5075|htgms          5075|ttgms          5075|NotMe             0
   rxcse            0|rxfme            0|rxlfm            0|timee             0
   LenEnd        5075|TimEnd           0|WaiEnd           0|E: enddl          3

Rx: 01 03 00 10 00 1C 45 C6 00 00 00 00 00 00 00 27 D5 00 00 00 00 00 00 00 00
Tx: 01 03 38 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Z   Zero Counters
```

In this shot, out of 5075 successful exchanges, there was none lost to errors. The last command received was 03 (Read Multiple Regs), starting at word 16, 28 words long.

NewProt	Protocol used. 1 or 2 for Modbus. See “Setting the GCM communications parameters” on page 2.
LastProt	Transient indication of the last protocol initialized. Normally same as “Prot”
BaudIdx	Index of the baud rate used: See “Setting the GCM communications parameters” on page 2.
LinePar	Line parameters. See “Setting the GCM communications parameters” on page 2.
ISRProt	Protocol Selector for the UART interrupt service routine. Typically same as Prot
Route	Routing for the XDC board. Always 0 for Modbus.
PrState	Prelude State. Not used, always zero.
Port	Port index. 0 for port 1 (UART 0, lower row on connector) 1 for port 2 (UART 1, upper row on connector)
Ints	UART Interrupt counter. Increments for any serial activity. Since the UARTs have 16 byte FIFOs, this counter typically increments slower than Rxchs and Txchs.
Rxchs	Received characters counter
Txchs	Transmitted characters counter
Rxuae	Receiver UART error counter.
Rxlua	UART Line Status Register at the last UART error. Bit encoded: Bit 0 Receiver data ready. This is not an error Bit 1 Overrun error Bit 2 Parity error Bit 3 Framing error Bit 4 Break condition Bit 5 Transmitter holding register empty. This is not an error Bit 6 Transmitter empty. This is not an error Bit 7 At least one error in receiver FIFO
Sta	UART Line status register at the last character received. See Rxlua above. Typically 0x61.
Cha	Hex representation of the last character received
Rxillmsr	Counter for illegal Modem Status Interrupt. The UART is configured to not generate MSRs. Only UART1 has modem status lines. Should always be 0.
Rxilluae	Counter for illegal (not enabled) UART error interrupts. Should always be 0.
Txboverr	Transmitter buffer overrun counter. Increments if an attempt is made to transmit characters past the end of the transmitter character buffer, indicating a bug in the protocol handler. Should always be 0.
Rxboverr	Receiver buffer overrun counter. Increments when the receiver buffer is filled up without an end of telegram condition.
Iir	UART Interrupt Identification Register at last interrupt.
Wastmo	Indicates that the last UART receiver interrupt was a UART character timeout (Stale data in FIFO) rather than a “FIFO trigger limit reached”.
Illints	Counter for undocumented IIR patterns. There appears to be a bug in the UART that, under adverse conditions, causes this to happen.
Unita	Controller/Station/Node number ID of the GCM in multidrop settings. Must be different for each connected GCM on the Modbus segment.
Map	Enumeration of the CIT map format. Set to 1 for standard CIT mapping.
MaxTime	Silence time for isTimeout (see below) in seconds. A value of zero is interpreted as “never” meaning isTimeout will never be true.

Rxsta	Character receiver state enumeration: 0 Waiting for first UART interrupt 1 Not used in Modbus 2 Buffering data 3 Not used in Modbus 4 Waiting for second end character (ModBus ASCII) 5 Received telegram complete on delimiters (ModBus ASCII) 6 Received telegram complete on UART timeout or intrinsic length (ModBus RTU) 7 Not used in Modbus 8 Not used in Modbus
Txsta	Character transmitter state enumeration: 0 Idle 1 First character in UART, RS422 transmitter enabled 2 Ongoing transmission from ISR 3 Last byte of telegram placed in UART FIFO 4 Waiting for UART transmitter empty to disable the RS422 transmitter
State	Communication task state enumeration: 0 Not used 1 Initializing (transient) 2 Waiting for some characters received 3 Waiting for end-of-telegram condition 4 Handling message 5 – 8 Not used in Modbus
isTimeout	No valid telegrams have been received for MaxTime (see above), and MaxTime is greater than zero.
addr	Starting Word in last CIT read or write telegram. Typically 16 or 44.
size	Number of Words in last CIT read or write telegram
cmd	Telegram command enumeration. Protocol dependent. For Modbus: 3 for Read Multiple Registers, 16 for Write Multiple Registers, 8 for Diagnostics
subf	ModBus telegram 8 subfunction. Only subfunction 0 (echo data) is supported.
cCRC	Calculated CRC in incoming telegram.
tCRC	CRC field in incoming telegram. Must be equal to cCRC for valid checksum.
bcnt	Byte count field in incoming and outgoing ModBus telegrams
lalen	Number of characters in the last completed incoming telegram
rtgms	Counter for incoming telegrams. This includes all telegrams with a valid end condition
htgms	Counter for handled telegrams. This includes all telegrams addressed to this GCM, correctly formatted, with a correct checksum.
ttgms	Counter for transmitted telegrams
NotMe	Counter for received, correctly formatted telegrams not addressed to this GCM
rxcse	Counter for received telegrams with checksum errors
rxfme	Counter for received, incorrectly formatted telegrams. See rxlfm below
rxlfm	Last encountered received format error enumeration. Protocol dependant. Note 1.
timee	Counter for isTimeout. Increments every time isTimeout goes from False to True
LenEnd	Counter for ModBus RTU telegrams who have had the end condition determined by the content of the length determining fields within the telegram
TimEnd	Counter for ModBus RTU telegrams who have had the end condition determined by UART character timeout

WaiEnd	Counter for ModBus RTU and UISServer telegrams who have had the end condition determined by absence of incoming characters for enddl (see below) 5 ms ticks.
enddl	Min wait in in 5 ms ticks for absence of incoming characters to determine end condition.

Note 1. Rxlfm indicates what kind of format error occurred last, as rxfme increments. In some cases, a NAK response is sent back to the master/client. This is indicated in the following table with a numerical value. An “x” indicates that the telegram will be ignored for the format error. An empty field indicates that the specific format error does not apply for this protocol.

Rxlfm	**ASCII**	**RTU**	**Format error**
1	x	x	Illegal receiver buffer length
2	x		Illegal characters in telegram – Modbus ASCII only
3			Not used in Modbus
4	3	3	Function 3 telegram not 8 bytes
5			Not used in Modbus
6	1	1	Unknown/Unsupported Function
7	3	3	Byte count disagrees with buffer length Function 16
8	3	3	Byte count disagrees with length field Function 16
9			Not used in Modbus
10	x		Illegal end sequence: LF not following CR – Modbus ASCII only
11	x	x	ISR Receiver buffer overrun
12	x	x	ISR Transmitter buffer overrun
13	x	x	ISR Characters received after end of message, before reply sent
14	x	x	ISR Illegal (default) receiver state (COMrxsta)
15	2	2	Outside CIT legal read space on CIT read or write
16	1	1	Unsupported subfunction for function 8 telegram
17	3	3	Function 8 telegram not 8 bytes
18	2	2	Write to read-only space in CIT
19-33			Not used in Modbus
34		x	Transmitter length rollover – Should never happen
35		x	Receiving characters while transmitting

Use the integrity bit.

In the PLC, toggle the integrity bits (Word 44, bit 7) every 2 seconds. Monitor the integrity echo bit (Word 16 bit 7). If they stop toggling, communications has failed, and appropriate steps can be taken. The integrity bit can be monitored in the TCMI CIT Diagnostic Screen (P1), and the GCM can be set up to monitor the activity of the integrity bit, potentially generating a Warning or Fault if the bit stops toggling.

PLC Program examples.

There are plenty of PLC example programs available in the Merrick MCT web site. See http://www2.merrick-inc.com/mct/.