RS-232 Communications

Series “C”, “CI”, “CM”, “IC3” and “T”

Users Manual

Myers Emergency Power Systems
44 South Commerce Way, Bethlehem, PA 18017
1-800-526-5088 • (610) 868-3500 • Fax: (610) 868-8686
Service: (610) 868-5400
www.myerseps.com

TABLE OF CONTENTS

1.0 - INTRODUCTION

This manual is intended to explain the RS-232 communication protocol for the “C”, “CI”, “CM” and “IC3” Emergency Lighting Central Inverter. Serial Communication can be established by means of a computer using terminal emulation software (such as Tera Term, PuTTy, xterm, etc.), or by any embedded device capable of RS-232 communication. We shall call this device the Client.

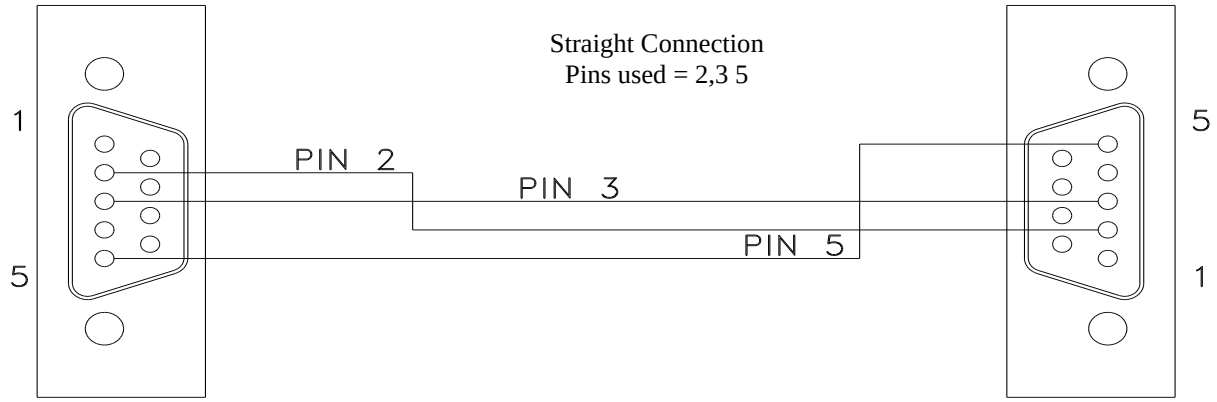
2.0 - CONNECTION

The “C”, “CI” and “IC3” Central Inverter has a 9-pin Sub-D (DB9) female connector located inside the inverter. See the Installation Guide for the exact location of the connector.
The “CM” and “T” Central Inverter has a 9-pin Sub-D (DB9) female connector located on the left-hand side of the inverter through a cutout.

The connection between the Client and the Inverter is a straight-through connection. Do ***not*** use a Null Modem Cable that flips pins 2 and 3. Pin 2 and Pin 3 are the Data send and receive lines; Pin 5 is the Ground (common) line. Optical isolation on the interface card provides galvanic isolation between the client device’s ground signal and the inverter’s ground.

Important: The RS-232 connection between the inverter and the Client is physically shared with the RS232 connection between the inverter and the front mounted ‘meter panel’ display. Since RS-232 only supports one device at a time, please disconnect the meter panel display before attempting RS-232 communication from the Client device. This can be done by opening the inverter cabinet door, and carefully disconnecting the connector on the back of the meter panel module that connects it with the inverter control board (which is deeper inside the cabinet).

Client Connector DB-9 **“C”, “CI”, “CM”, “IC3” & “T” Central Inverter Connector DB-9**



3.0 - TERMINAL SETTINGS

Communication is established through a standard ASCII format of 8 Data bits, 1 Stop bit, No parity, No Flow Control, and a Baud rate of 9600 BPS.

Baud Rate:	9,600
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None
Character Set:	ANSI

The RS-232 protocol uses carriage returns (‘\r’, ANSI code 0x0D), but does ***not*** use line feeds (‘\n’, ANSI code 0x0A). If you are manually typing commands, some terminal emulator software will automatically feed the line back when you press enter (and a carriage return is sent) and when the inverter responds (with a response string followed by a carriage return). However, some terminal emulator software will not do this, and you will end up with something like the following where each line is staggered:

```
*0DA0
     *0da09/16/19X
                  *0TM0
                       *0tm12:59X
```

If this is the case, look for the setting in your terminal emulator software that automatically appends line feeds after (outgoing or incoming) carriage returns.

Finally, the inverter will not echo back characters you type. If it looks like nothing is happening when you type, until you press enter, then turn on the ‘local echo’ feature in your terminal emulator software.

4.0 - PROTOCOL OVERVIEW

The RS-232 protocol specifies the ‘language’ for commands and responses between the Client and the inverter. Information is exchanged between the two devices in this ‘language’.

The Client is the initiator of all communications. The Client sends out commands in all upper-case letters, like the following string. Commands must be typed perfectly; The inverter will just drop (ignore) mistyped commands – you will not receive any error feedback. You may use the backspace key to fix typos. The square brackets (‘[‘ and ‘]’) below are not transmitted in the protocol; they are just in this document for clarity, to help differentiate fields that are dynamic (as opposed to fields that are static, or fixed).

***ACC**[**DDDD**] [**P**]<CR>

*****	Start Character, marks the start of a command.
A	Character represents the System’s node address. This will always be 0.
CC	Two characters that specify the command (upper case!).
DDDD	Data Field with variable length (often there is no data and the length is zero).
P	Phase (0, 1 or 2). This is almost always 0, but for commands that can be addressed to each of the three phases of a 3-phase inverter, you would use 0 for phase A, 1 for phase B and 2 for phase C.
<CR>	Carriage return character (‘\r’, ANSI code 0x0d).

When the inverter correctly receives this string of characters, it will reply. The inverter usually (not always) sends back the same command but in lower case letters, and in the following syntax:

***acc**[**dddd**] [**uuuu**]**t**<CR>

*****	Start Character, marks the start of a command.
a	Character represents the System’s node address. This will always be 0.
cc	Two characters that reflect the command that is being responded to (lower case!).
dddd	Data Field with variable length (sometimes there is no data and the length is zero).
uuuu	The ‘Unit’ field, with variable length. For example, ‘V’ for volts, ‘Min’ for minutes, etc. For 3-phase inverters, this may also contain the phase for which the data was returned.
t	Termination character. Either ‘0’ or ‘X’.
<CR>	Carriage return character (‘\r’, ANSI code 0x0d).

5.0 - COMMANDS

Meter Functions

5.1 Get Alarm Status

Command: ***0AS0**<CR>
Response: ***0as[dddd]0**<CR>

dddd represents a 16-bit field in HEX format. Please refer to the table below for bit definitions.

Bit	Alarm Name	Alarm Description
0	Overload Alarm	Too much load connected to inverter output
1	Overload Shutdown Alarm	Inverter has shutdown due to too much load
2	High AC Voltage Alarm	Excessively high input voltage
3	Low AC Voltage Alarm	Excessively low input voltage
4	High Ambient Alarm	Excessively high ambient temperature
5	Circuit Breaker Alarm	Circuit breaker(s) has/have tripped
6	Load Reduction Alarm	Load has reduced below programmed threshold (a load device has gone offline – perhaps a lamp is out?)
7	Near Low Battery Alarm	Inverter is approaching low battery
8	Low Battery Alarm	Inverter battery voltage is critically low
9	Inverter Failure Alarm	Inverter cannot output voltage (problem with circuitry)
10	Charge Failure Alarm	The battery charging circuitry has failed
11	Utility Power Down Alarm	No utility power; inverter is running on battery
12	Spare	**reserved**
13	Output Fault Alarm	There is an issue with the load (such as a short circuit)
14	Spare	**reserved**
15	Spare	**reserved**

The conversion of each of the four hex characters (in **dddd**) from hex to binary is as follows:

Hex	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

If there were an overload alarm, the Alarm status would read 0001 in hex, which would be 0000 0000 0000 0001 in binary. As can be seen, the last bit (bit 0) is set, indicating an overload alarm.

Example:
Command: ***0AS0**<CR>
Response: ***0as[088B]0**<CR>

In this theoretical response, 0x088B is translated into binary as 0000 1000 1000 1011 (where the leftmost bit is bit15, and the rightmost bit is bit0), which tells us that the following alarm conditions are asserted: Overload, Overload Shutdown, Low AC Voltage, Near Low Battery, Utility Power Down.

5.2 Get Indicator Status

Command: ***0IS0**<CR>
Response: ***0is[dddd]0**<CR>
dddd represents a 16-bit field in HEX format. Please refer to the table below for bit definitions.

Bit	Status Flag Name	Status Flag Description
0	System Ready	0 while rebooting, on battery or charging. 1 all other times.
1	AC Present	0 when there is no AC input voltage from the utility (i.e. utility power outage). 1 otherwise.
2	Battery Charging	0 when on battery power, 1 otherwise (the battery is always charging when power is not being drawn from it. When its full, the inverter uses 'float charging' to keep it full)
3	On Battery Power	1 when the inverter is running on battery power. 0 otherwise.
4	Spare	**reserved**
5	Spare	**reserved**
6	Spare	**reserved**
7	Is 3-phase inverter	0 if the inverter is single-phase. 1 if the inverter is three-phase.
8	Spare	**reserved**
9	Spare	**reserved**
10	Spare	**reserved**
11	Spare	**reserved**
12	Spare	**reserved**
13	Spare	**reserved**
14	Spare	**reserved**
15	Spare	**reserved**

This format is just like the Alarm status. Please read Hex format from Alarm Status (above).

5.3 Get Input Voltage

Command: ***0VI[ϕ]**<CR>
Where ϕ represents the phase. Use '**0**' for phase A, '**1**' for phase B or '**2**' for phase C.
For a single-phase inverter, always use '**0**')

Response: ***0vi[ddd.d]VX**<CR> (for a single-phase inverter)
***0vi[ddd.d]V òAX**<CR> (for phase A on 3-phase inverter)
***0vi[ddd.d]V òBX**<CR> (for phase B on 3-phase inverter)
***0vi[ddd.d]V òCX**<CR> (for phase C on 3-phase inverter)
Where **ddd.d** is the voltage in Volts (for example, **120.6**). The character '**ò**' (ANSI code 0xF2) means 'phase' (there is no 8-bit ANSI code for 'ϕ' - the Greek letter Phi that is usually used to mean 'phase' – in the US ANSI-8 character set).

Example 1: You query a single-phase inverter whose input voltage is 118.6 Volts:
(you send) ***0VI0**<CR>
(it responds) ***0vi118.6VX**<CR>
Example 2: You query a three-phase inverter whose input voltages are 278.6 Volts (phase A), 277.9 Volts (phase B) and 277.6 Volts (phase C):

(you send) ***0VI0**<CR>
(it responds) ***0vi278.6V òAX**<CR>
(you send) ***0VI1**<CR>
(it responds) ***0vi277.9V òBX**<CR>
(you send) ***0VI2**<CR>
(it responds) ***0vi277.6V òCX**<CR>

5.4 Get Output Voltage

Command: ***0VO**[ϕ]<CR>
Where ϕ represents the phase. Use '**0**' for phase A, '**1**' for phase B or '**2**' for phase C. For a single-phase inverter, always use '**0**')

Response: ***0vo**[**ddd.d**]**VX**<CR> (for a single-phase inverter)
***0vo**[**ddd.d**]**V òAX**<CR> (for phase A on 3-phase inverter)
***0vo**[**ddd.d**]**V òBX**<CR> (for phase B on 3-phase inverter)
***0vo**[**ddd.d**]**V òCX**<CR> (for phase C on 3-phase inverter)

Where **ddd.d** is the voltage in Volts (for example, **121.2**). For more information and to understand the 3-phase syntax better, see the comments and read the examples in section 5.3 – Get Input Voltage.

5.5 Get Output Current

Command: ***0IO**[ϕ]<CR>
Where ϕ represents the phase. Use '**0**' for phase A, '**1**' for phase B or '**2**' for phase C. For a single-phase inverter, always use '**0**')

Response: ***0io**[**ddd.d**]**AX**<CR> (for a single-phase inverter)
***0io**[**ddd.d**]**A òAX**<CR> (for phase A on 3-phase inverter)
***0io**[**ddd.d**]**A òBX**<CR> (for phase B on 3-phase inverter)
***0io**[**ddd.d**]**A òCX**<CR> (for phase C on 3-phase inverter)

Where **ddd.d** is the current in Amps (for example, **.7**, **8.2**, **23.8**, or **148.8**). For more information and to understand the 3-phase syntax better, see the comments and read the examples in section 5.3 – Get Input Voltage.

5.6 Get Battery Voltage

Command: ***0BV0**<CR>
Response: ***0bv**[**ddd.d**]**VX**<CR>

Where **ddd.d** is the voltage in Volts (for example, **48.6**, **132.5**, or **267.2**).

5.7 Get Battery Current

Command: ***0BI0**<CR>
Response: ***0bi**[**ddd.d**]**AX**<CR>

Where **ddd.d** is the current in Amps (for example, **.1**, **5.6**, **31.3**, or **104.7**). Note that when current is less than 1A, the leading zero is not displayed (i.e. **.1**, not **0.1**).

5.8 Get Ambient Temperature

Command: ***0TP0**<CR>
Response: ***0tp**[**ddd.d**]**ßCX**<CR>

Where **ddd.d** represents the temperature in degrees Celsius (for example, **26.6**, **30.8**, or **33.0**). The Beta symbol ('ß', ANSI code 0xDF) can be ignored. To convert to degrees Fahrenheit, subtract 32, then multiply by 5, then divide by 9. $°F = \frac{5(°C-32)}{9}$

5.9 Get Battery Output Power

Command: ***0WA0**<CR>
Response: ***0wa**[**dddd**]**X**<CR>

Where dddd represents the output power from the batteries, in Watts (for example, **27**, **8045**, **19822**, or **44781**). Note that there is no decimal point; the output is a whole number in Watts. When on utility power, this value will be very low.

5.10 Get Output VA

Command: ***0VA**[**ϕ**]<CR>
Where ϕ represents the phase. Use '**0**' for phase A, '**1**' for phase B or '**2**' for phase C. For a single-phase inverter, always use '**0**')

Response:	***0va**[**dddd**]**VAX**<CR>	(for a single-phase inverter)
	***0va**[**dddd**]**VA òAX**<CR>	(for phase A on 3-phase inverter)
	***0va**[**dddd**]**VA òBX**<CR>	(for phase B on 3-phase inverter)
	***0va**[**dddd**]**VA òCX**<CR>	(for phase C on 3-phase inverter)

Where **dddd** is the output power of that phase of the inverter in VA (Vars – or Voltage-Ampere Reactive). For example, **6824**, **13219**, or **21688**). Note that there is no decimal point; the output is a whole number in VA. For more information and to understand the 3-phase syntax better, see the comments and read the examples in section 5.3 – Get Input Voltage.

5.11 Get Elapsed System-Up Time (days)

Command: ***0ED0**<CR>
Response: ***0ed**[**dddd**]**X**<CR>

Where **dddd** represents the number of days the inverter has been up and running (for example, **9**, **62**, **113**, or **2128**).

5.12 Get Inverter Run time

Command: ***0RT0**<CR>
Response: ***0et**[**dddd**]**MinX**<CR>

Where **dddd** represents the number of minutes that the inverter has run the output on battery power (for example, **9**, **62**, **113**, or **2128**).
IMPORTANT NOTE: This is one of the few commands where the response from the inverter ("**et**") has different letters than the command ("**RT**").

Control Functions

5.13 Set Date

Command: ***0DA[MM][DD][YY]0**<CR>
Where **MM** is month (01=January...12=December), **DD** is day (01..31) and **YY** is year (**19** = 2019). Do not forget the final **0** in the command, or it won't work.
Response: None! Use the Get Date command to confirm it worked.

5.14 Get Date

Command: ***0DA0**<CR>
Response: ***0da[mm][dd][yy]X**<CR>
Where **mm** is month (01=January...12=December), **dd** is day (01..31) and **yy** is year (**19** = 2019)

5.15 Set Time

Command: ***0TM[HH][MM]0**<CR>
Where **HH** is the hour in 24-hour format (00..23), and **MM** is the minute (00..59)
Response: None! Use the Get Time command to confirm it worked.

5.16 Get Time

Command: ***0TM0**<CR>
Response: ***0tm[hh]:[mm]X**<CR>
Where **hh** is the hour in 24-hour format (00..23), and **mm** is the minute (00..59)

5.17 Set Inverter Max Run Time

Command: ***0MT[HHHH]0**<CR>
Where **HHHH** is the maximum 'battery power' runtime that you want to allow the inverter to reach (in hours).
Response: None! Use the Get Inverter Max Run Time command to confirm it worked

5.18 Get Inverter Max Run Time

Command: ***0MT0**<CR>
Response: ***0mt[hhhh]X**<CR>
Where **hhhh** is the maximum 'battery power' runtime (in hours) that the inverter is allowed to reach.

5.19 Set Output Current Load Reduction Fault

Command: ***0F[ϕ][DDDD]0**<CR>
ϕ is the phase (**A** for phase A or if single-phase, **B** for phase B, or **C** for phase C). **DDDD** is the 'nominal output current' (in tenths of Amps. For example, to set 123.4 Amps, enter **1234.** To turn off this feature, use **0000**) that you consider normal (all loads functioning normally). If you set this value, and if the output current dips below 90% of this value, the Load Reduction Fault alarm will trigger. Do not forget the final 0 to terminate the command, or it won't work properly.
Response: None! Use the Get Output Current Load Reduction Fault command to confirm it worked.

5.20 Get Output Current Load Reduction Fault

Command: ***0F[ϕ]0**<CR>

ϕ is the phase (**A** for phase A or if single-phase, **B** for phase B, or **C** for phase C).

Response: ***0lf[ddd.d]AX**<CR>

dddd is the 'nominal output current' (in Amps). If this value is set, and if the current value of output current dips below 90% of this value, the Load Reduction Fault alarm will trigger. Note that leading zeros are not displayed for currents below 1A (e.g. **.4** for 0.4A). If the feature is off, **.0** will be returned.

5.21 Set Low Battery Voltage Alarm

Command: ***0LB[DDDD]0**<CR>

Where **DDDD** is the voltage (in tenths of a Volt, for example 216.5 Volts would be entered as **2165**) that you wish to set as the 'low battery voltage' threshold that triggers the Low Battery Voltage Alarm. Do not forget the final 0 to terminate the command, or it won't work properly.

Response: None! Use the Get Low Battery Voltage Alarm command to confirm it worked.

5.22 Get Low Battery Voltage Alarm

Command: ***0LB0**<CR>

Response: ***0lb[ddd.d]VX**<CR>

Where **ddd.d** is the voltage (in Volts) that is currently set as the 'low battery voltage' threshold that triggers the 'Low Battery Voltage' alarm.

5.23 Set Low AC Voltage Alarm

Command: ***0LV[DDDD]0**<CR>

Where **DDDD** is the voltage (in tenths of a Volt, for example 250.5 Volts would be entered as **2505**) that you wish to set as the 'low AC input voltage' threshold that triggers the 'Low AC Voltage' alarm.

Response: None! Use the Get Low AC Voltage Alarm command to confirm it worked.

5.24 Get Low AC Voltage Alarm

Command: ***0LV0**<CR>

Response: ***0lv[ddd.d]VX**<CR>

Where **ddd.d** is the voltage (in Volts) that is currently set as the 'low AC input voltage' threshold that triggers the 'Low AC Voltage' alarm.

5.25 Set High AC Voltage Alarm

Command: ***0HV[DDDD]0**<CR>

Where **DDDD** is the voltage (in tenths of a Volt, for example 318.6 Volts would be entered as **3186**) that you wish to set as the 'high AC input voltage' threshold that triggers the 'High AC Voltage' alarm.

Response: None! Use the Get High AC Voltage Alarm command to confirm it worked.

5.26 Get High AC Voltage Alarm

Command: ***0HV0**<CR>
Response: ***0hv[ddd.d]VX**<CR>
Where **ddd.d** is the voltage (in Volts) that is currently set as the 'high AC input voltage' threshold that triggers the 'High AC Voltage' alarm.

5.27 Set Ambient Temperature Alarm

Command: ***0AT[DD]0**<CR>
Where **DD** is temperature (in degrees Celsius) that you wish to set as the 'high ambient temperature' threshold that triggers the 'High Ambient' alarm. Note: to translate from degrees Fahrenheit to Celsius, multiply by 9, divide by 5, then add 32. $°C = \frac{9(°F)}{5} + 32$
Response: None! Use the Get Ambient Temperature Alarm command to confirm it worked.

5.28 Get Ambient Temperature Alarm

Command: ***0AT0**<CR>
Response: ***0at[dd]CX**<CR>
Where **dd** represents the temperature (in degrees Celsius) that is currently set as the 'high ambient temperature' threshold that triggers the 'High Ambient' alarm. To convert to degrees Fahrenheit, subtract 32, then multiply by 5, then divide by 9. $°F = \frac{5(°C-32)}{9}$

Log Functions

5.29 Get Number of Test Log Entries

Command: ***0LT9100**<CR>

If there are no self-tests in the log yet:
Response: ***0ltNo Tests X**<CR>

If there is at least one self-test in the log, or more:
Response: ***0ltT[dd]/[ee]X**<CR>
Where **dd** is the last log entry that was fetched (ignore this value), and **ee** is the current 'total number of Test Log entries' (max possible value is 75).

5.30 Get Test Log Entry Field

Each time the inverter runs an automatic self-test, it logs the results in the Test Log as multiple fields. Each Test Log entry has 12 fields (numbered **00** through **11**) that can be individually retrieved.

Command: ***0LT[ii][jj]**<CR>
Where **ii** is the Test Log entry number (must be a two-digit number between **01** and the 'number of Test Log entries' (see section 5.29 above), and **jj** is the field number (must be a two-digit number between **00** and **11**).

See the table below for the field number specifications, and the type of response that is generated with each field number.

Field #	Field name	Response Syntax	Description
00	Date	*0lt[mm]/[dd]/[yy]X<CR>	mm is month (01..12), dd is day (01..31), yy is year
01	Time	*0lt[hh]:[mm]X<CR>	hh is hour (00..23, 24-hour format) and mm is minute (00..59)
02	Duration	*0lt[dd]MinX<CR>	dd is the test duration, in minutes (for example, 5, 20, etc)
03	Output Voltage Phase A	*0lt[ddd.d]V òAX<CR>	ddd.d is the single-phase or Phase A output voltage when the test was run, in Volts.
04	Output Voltage Phase B	*0lt[ddd.d]V òBX<CR>	ddd.d is the Phase B output voltage when the test was run, in Volts. Only applicable if 3-phase inverter.
05	Output Voltage Phase C	*0lt[ddd.d]V òCX<CR>	ddd.d is the Phase C output voltage when the test was run, in Volts. Only applicable if 3-phase inverter.
06	Output Current Phase A	*0lt[ddd.d]A òAX<CR>	ddd.d is the single-phase or Phase A output current when the test was run, in Amps
07	Output Current Phase B	*0lt[ddd.d]A òBX<CR>	ddd.d is the Phase B output current when the test was run, in Amps. Only applicable if 3-phase inverter.
08	Output Current Phase C	*0lt[ddd.d]A òCX<CR>	ddd.d is the Phase C output current when the test was run, in Amps. Only applicable if 3-phase inverter.
09	Ambient Temp	*0lt[dd.d]ßCX<CR>	dd.d is the ambient temperature when the test was run, in degrees Celsius
10	Alarm Status	*0ltAlarms: NoX<CR> *0ltAlarms: YesX<CR>	Indicates whether or not any alarm conditions were asserted at the time in which the self-test was run
11	Test Trigger	*0ltMonthlyX<CR> *0ltYearlyX<CR>	Indicates what triggered the self-test (either it was an automatic monthly self-test, or an automatic yearly one)

***Example**:* In the example, we retrieve the third self-test log. Let's assume this test was run on February 16th 2018, at 11:30pm, it took 5 minutes to complete, the output voltages on the three phases were 277.1V, 277.2V and 276.9V respectively, the output currents on the three phases were 28.1A, 27.3A and 29.7A respectively, the ambient temperature was 24.6°C, no alarms were asserted at the time of the test, and the test was an automatic monthly self-test.

(you send) ***0LT0300<CR>**
(it responds) ***0lt02/16/18X<CR>**

(you send) ***0LT0301<CR>**
(it responds) ***0lt23:30X<CR>**

(you send) ***0LT0302<CR>**
(it responds) ***0lt5MinX<CR>**

(you send) ***0LT0303<CR>**
(it responds) ***0lt277.1V òAX<CR>**

(you send) ***0LT0304<CR>**
(it responds) ***0lt277.2V òBX<CR>**

(you send) ***0LT0305<CR>**
(it responds) ***0lt276.9V òCX<CR>**

(you send) ***0LT0306<CR>**
(it responds) ***0lt28.1A òAX<CR>**

(you send) ***0LT0307<CR>**
(it responds) ***0lt27.3A òBX<CR>**

(you send) ***0LT0308**<CR>
(it responds) ***0lt29.7A òCX**<CR>

(you send) ***0LT0309**<CR>
(it responds) ***0lt24.6ßCX**<CR>

(you send) ***0LT0310**<CR>
(it responds) ***0ltAlarms: NoX**<CR>

(you send) ***0LT0311**<CR>
(it responds) ***0ltMonthlyX**<CR>

5.31 Dump Test Log

Command: ***0DT0**<CR>

This command dumps all Test Log entries to the RS-232 port. It is essentially a script which reads out all log entries (where for each log entry, it reads out all fields). Each field gets its own line. Log entries are separated by two blank lines. Here is a sample of the response syntax (showing the first two log entries, and the last log entry):

(you send) ***0DT0**
(it responds...)

```
*0lt10/15/15X
*0lt05:00X
*0lt5MinX
*0lt283.1V òAX
*0lt283.2V òBX
*0lt281.9V òCX
*0lt8.6A òAX
*0lt12.6A òBX
*0lt9.1A òCX
*0lt30.0ßCX
*0ltAlarms: NoX
*0ltMonthlyX


*0lt11/15/15X
*0lt05:00X
*0lt5MinX
*0lt283.6V òAX
*0lt282.1V òBX
*0lt282.6V òCX
*0lt8.7A òAX
*0lt12.8A òBX
*0lt9.2A òCX
*0lt27.2ßCX
*0ltAlarms: NoX
*0ltMonthlyX

  :   :   :

*0lt10/15/19X
*0lt05:00X
*0lt5MinX
*0lt284.2V òAX
*0lt282.8V òBX
*0lt282.8V òCX
*0lt8.6A òAX
*0lt11.7A òBX
*0lt9.3A òCX
*0lt31.1ßCX
*0ltAlarms: NoX
*0ltMonthlyX
```

5.32 Get Number of Event Log Entries

Command: ***0LE**9100<CR>

If there are no manually invoked tests (AKA 'events') in the log yet:
Response: ***0leNo EventsX**<CR>

If there is at least one event in the log, or more:
Response: ***0leE**[**dd**]**/**[**ee**]**X**<CR>
Where **dd** is the last log entry that was fetched (ignore this value), and **ee** is the current 'total number of Event Log entries' (max possible value is 75).

5.33 Get Event Log Entry Field

Each time a user manually invokes an inverter self-test (AKA 'event'), the inverter logs the results in the Event Log as multiple fields. Each Event Log entry has 11 fields that can be individually retrieved.

Command: ***0LE**[**ii**][**jj**]<CR>
Where **ii** is the Event Log entry number (must be a two-digit number between **01** and the 'number of Event Log entries' (see section 5.32 above), and **jj** is the field number (must be a two-digit number between **00** and **10**).

The fields in the Event Log are exactly the same as the fields in the Test Log, except that the last field ('Test Trigger') does not exist in an Event Log entry (in other words, there is one fewer field).
See the table in section 5.30 above for a description of the fields.

5.34 Dump Event Log

Command: ***0DE0**<CR>
This command dumps all Event Log entries to the RS-232 port. It is essentially a script which reads out all log entries (where for each log entry, it reads out all fields). Each field gets its own line. Log entries are separated by two blank lines. Here is a sample of the response syntax (showing the first two log entries, and the last log entry):

(you send) ***0DE0**
(it responds...)

```
*0le02/03/18X
*0le04:59X
*0le1MinX
*0le283.3V òAX
*0le284.6V òBX
*0le282.0V òCX
*0le8.0A òAX
*0le11.6A òBX
*0le8.9A òCX
*0le34.3ßCX
*0leAlarms: NoX


*0le07/31/18X
*0le06:04X
*0le1MinX
*0le283.9V òAX
*0le282.9V òBX
*0le282.4V òCX
*0le8.1A òAX
*0le12.7A òBX
*0le9.1A òCX
*0le34.1ßCX
*0leAlarms: NoX
  :   :   :
```

```
*01e09/02/19X
*01e19:32X
*01e1MinX
*01e284.7V òAX
*01e282.5V òBX
*01e295.3V òCX
*01e7.3A òAX
*01e11.4A òBX
*01e.4A òCX
*01e34.8ßCX
*01eAlarms: NoX
```

5.35 Get Number of Alarm Log Entries

Command: ***0AL9100**<CR>

If there are no alarms logged in the Alarm Log yet:
Response: ***0alNo Alarms X**<CR>

If there is at least one alarm log entry, or more:
Response: ***0alA[dd]/[ee]X**<CR>

Where **dd** is the last log entry that was fetched (ignore this value), and **ee** is the current 'total number of Alarm Log entries' (max possible value is 50).

5.36 Get Alarm Log Entry Field

Each time a new alarm condition asserts, the inverter logs the alarm in the Alarm Log as multiple fields. Each Alarm Log entry has 3 fields that can be individually retrieved.

Command: ***0AL[ii][jj]**<CR>

Where **ii** is the Alarm Log entry number (must be a two-digit number between **01** and the 'number of Alarm Log entries' (see section 5.35 above), and **jj** is the field number (must be a two-digit number between **00** and **02**).

See the table below for the field number specifications, and the type of response that is generated with each field number.

Field #	Field name	Response Syntax	Description
00	Date	***0al[mm]/[dd]/[yy]X**<CR>	**mm** is month (01..12), **dd** is day (01..31), **yy** is year
01	Time	***0al[hh]:[mm]X**<CR>	**hh** is hour (00..23, 24-hour format) and **mm** is minute (00..59)
02	Alarm	***0al[ssss]X**<CR>	**ssss** is a text string that describes the alarm. The possible values of the string are: • **Overload** • **OverloadShutdown** • **High AC Voltage** • **Low AC Voltage** • **High Ambient** • **Circuit Breaker** • **Load Variation** (Load Reduction) • **Near Low Battery** • **Low Battery** • **Inverter** • **Charger** • **Utility Failure** • **Output** These correspond with the alarm statuses described in section 5.1 above.

***Example**:* In the example, we retrieve the third Alarm Log entry. Let's assume the alarm triggered on November 8th, 2017, at 3:06pm, and the alarm was a high AC voltage.

(you send) ***0AL0300**<CR>
(it responds) ***0al11/08/17X**<CR>

(you send) ***0AL0301**<CR>
(it responds) ***0al15:06X**<CR>

(you send) ***0AL0302**<CR>
(it responds) ***0alHigh AC VoltageX**<CR>

5.37 Dump Alarm Log

Command: ***0DF0**<CR>

This command dumps all Alarm Log entries to the RS-232 port. It is essentially a script which reads out all log entries (where for each log entry, it reads out all fields). Each field gets its own line. Log entries are separated by two blank lines. Here is a sample of the response syntax (showing the first two log entries, and the last log entry):

(you send) ***0DF0**
(it responds…)

```
*0al09/30/18X
*0al05:52X
*0alUtility FailureX


*0al01/03/19X
*0al03:02X
*0alLoad VariationX

   :   :   :    :

*0al09/26/19X
*0al14:54X
*0alUtility FailureX
```

Self-Test Functions

5.38 Initiate Self-Test

Command: ***0TS0**<CR>
Response: ***0ts0**<CR>

This command invokes a manual inverter self-test, testing the battery power and inverter operation. The Response (***0ts0**<CR>) gets sent immediately on receipt of the command. Once the self-test completes, there is no notification over RS-232. The results of the test will be stored in the Event Log (see sections 5.32 through 5.34 above).

5.39 Set Monthly Self-Test Day

Command: ***0MD**[**DD**]**0**<CR>
Where **DD** is the day of the month (01..28) on which to run the monthly inverter self-test (it is not recommended to use 29, 30 or 31 as the day, as not all months have those days). Do not forget the final **0** in the command, or it won't work.
Response: None! Use the Get Monthly Self-Test Day command to confirm it worked.

5.40 Get Monthly Self-Test Day

Command: ***0MD0**<CR>
Response: ***0md[dd]X**<CR>
Where **dd** is the day of the month (01..31) in which the monthly self-test will be run.

5.41 Set Monthly Self-Test Time

Command: ***0TT[HH][MM]0**<CR>
Where **HH** is the hour in 24-hour format (00..23), and **MM** is the minute (00..59) at which you would like the monthly self-test to run when the programmed 'day of the month' is reached. It is recommended to use a time where it is unlikely that people are around the inverter, as it will make a humming noise while on battery power. Do not forget the final **0** in the command, or it won't work.
Response: None! Use the Get Monthly Self-Test Time command to confirm it worked.

5.42 Get Monthly Self-Test Time

Command: ***0TT0**<CR>
Response: ***0tt[hh][mm]X**<CR>
Where **hh** is the hour in 24-hour format (00..23), and **mm** is the minute (00..59) at which the monthly self-test will be run when the programmed 'day of the month' is reached.

5.43 Set Yearly Self-Test Month and Day

Command: ***0YD[MM][DD]0**<CR>
Where **MM** is the month (01..12) and **DD** is the day of the month (01..31) on which to run the yearly inverter self-test. Do not forget the final **0** in the command, or it won't work.
Response: None! Use the Get Yearly Self-Test Month and Day command to confirm it worked.

5.44 Get Yearly Self-Test Month and Day

Command: ***0YD0**<CR>
Response: ***0yd[mm][dd]X**<CR>
Where **mm** is the month (01..12) and **dd** is the day of the month (01..31) on which the yearly self-test will be run.

5.45 Set Yearly Self-Test Time

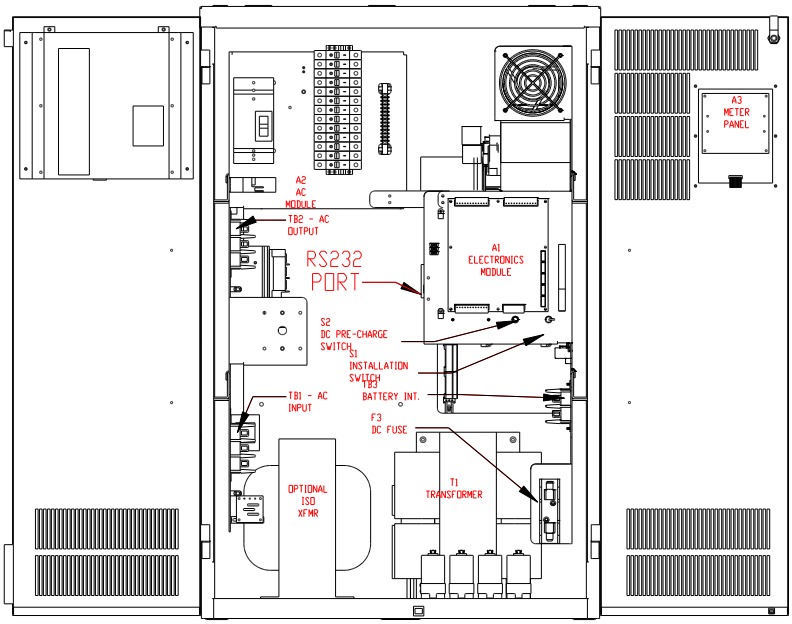
Command: ***0YT[HH][MM]0**<CR>
Where **HH** is the hour in 24-hour format (00..23), and **MM** is the minute (00..59) at which you would like the yearly self-test to run when the programmed month and day are reached. It is recommended to use a time where it is unlikely that people are around the inverter, as it will make a humming noise while on battery power. Do not forget the final **0** in the command, or it won't work.
Response: None! Use the Get Yearly Self-Test Time command to confirm it worked.

5.46 Get Yearly Self-Test Time

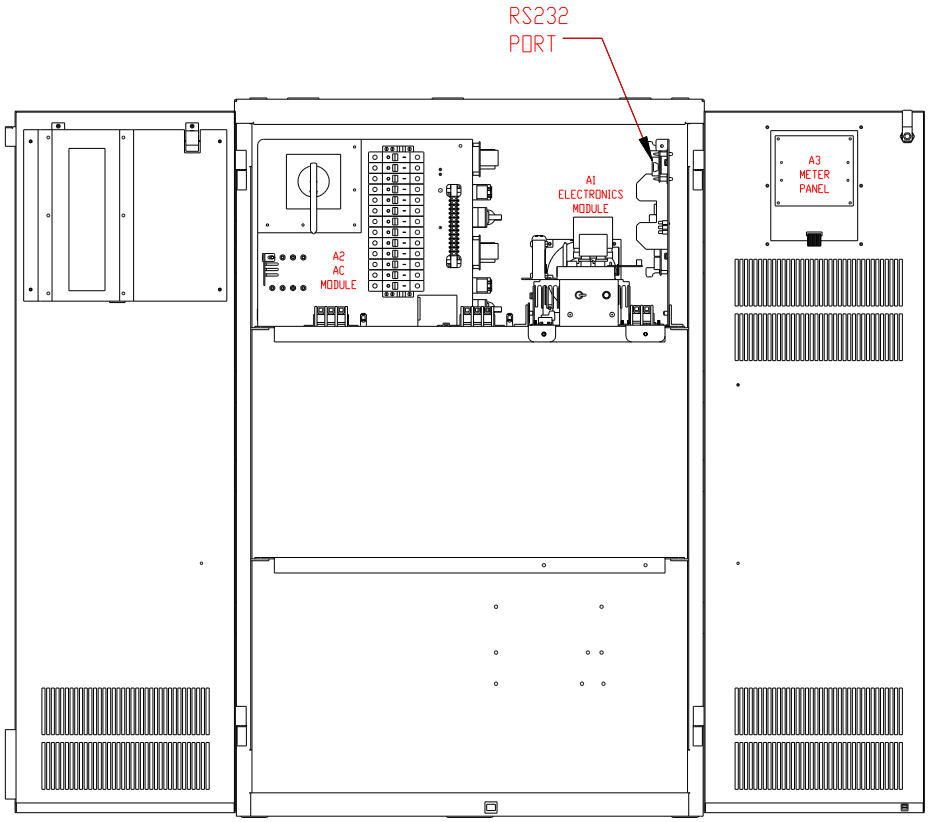
Command: ***0YT0**<CR>
Response: ***0yt[hh][mm]X**<CR>
Where **hh** is the hour in 24-hour format (00..23), and **mm** is the minute (00..59) at which the yearly self-test will be run when the programmed month and day are reached.

6.0 - LOCATING THE RS-232 PORT

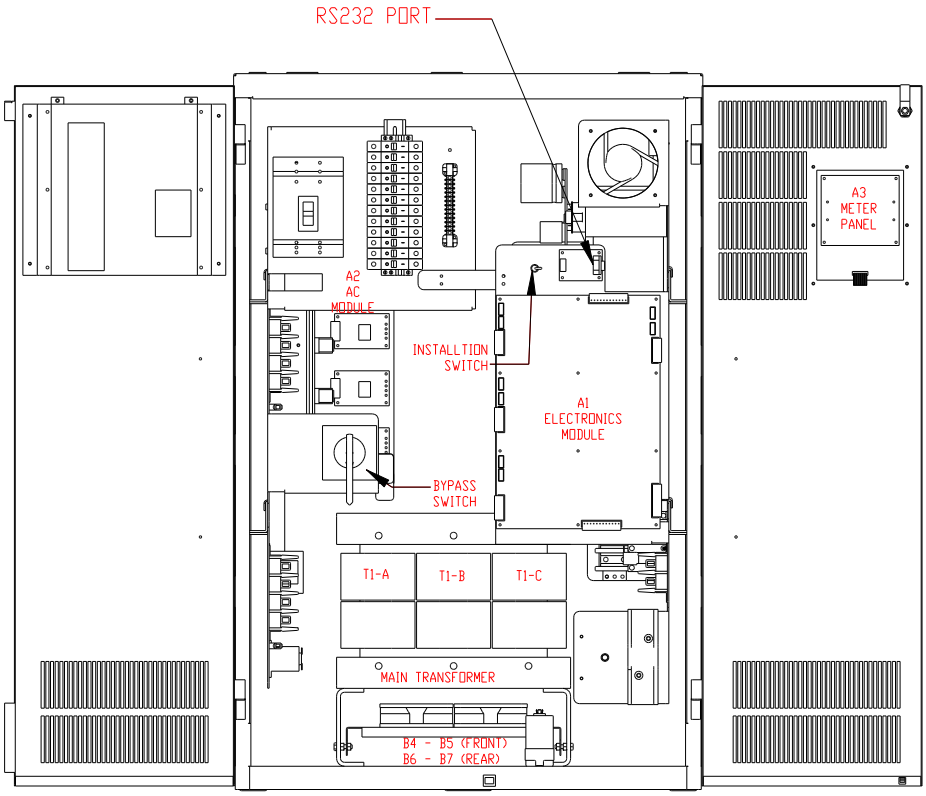
LOCATION OF THE RS-232 PORT FOR "C"& "CI" 6-16.7k



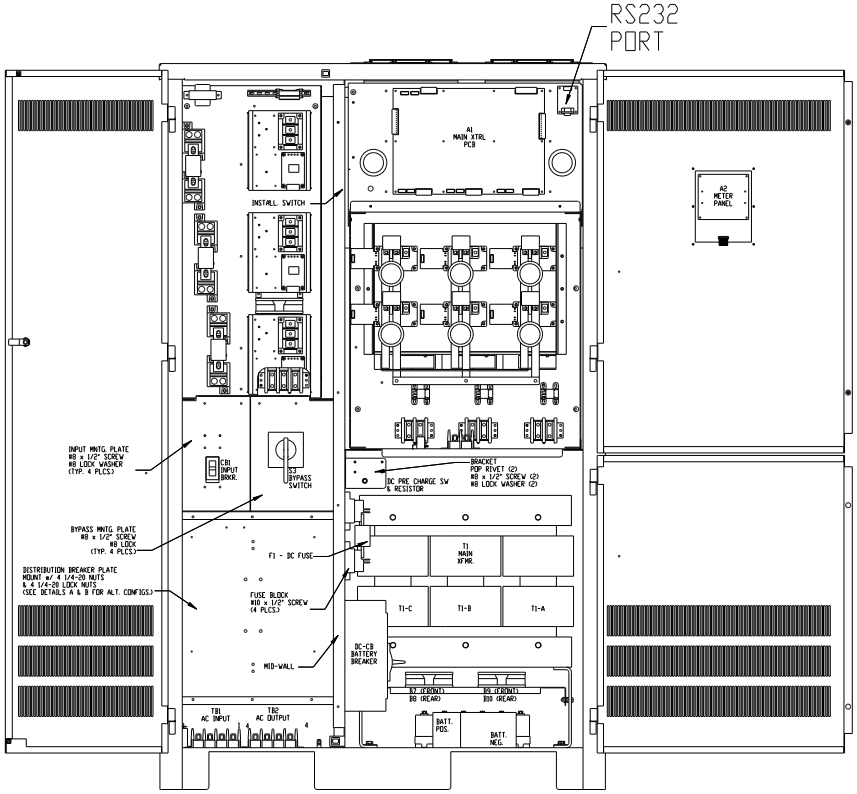
LOCATION OF THE RS-232 PORT FOR "C" & "CI" 1.25 –4.8k



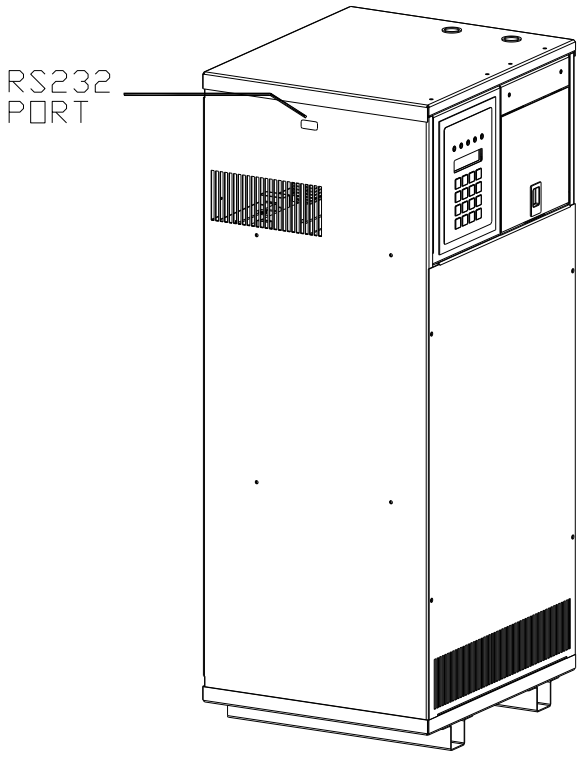
LOCATION OF THE RS-232 PORT FOR "IC3" 4.8 –16.7k



LOCATION OF THE RS-232 PORT FOR "IC3" 24-50kVA



LOCATION OF THE RS-232 PORT FOR "T" 750W – 1350W



7.0 - THE ZOOM MODEM (OPTIONAL)

Configuring the Zoom Modem

Connect the 9VDC Power Adapter
Connect the PC serial port to the modem's serial port (if your PC doesn't have a serial port, you can use an off-the-shelf 'USB to RS-232' adapter.
On the PC, bring up a terminal emulator program such as PuTTy, Tera Term or xterm.
Configure the terminal emulator to the following:

9600 BPS
8 Data Bits
No Parity
No Stop Bits

No Flow Control
Make sure there is communication by typing AT<enter> until the message "OK" appears.
Type the following AT commands:

ATM1 <enter>	(speaker on until connected)
AT&D0 <enter>	(ignore DTR)
AT&K0 <enter>	(no flow control)
ATS0=1 <enter>	(auto-answer after one ring)
AT&W0	(store to non-volatile memory)

Dialing the ZOOM Modem

Type AT<enter> until the "OK" message appears

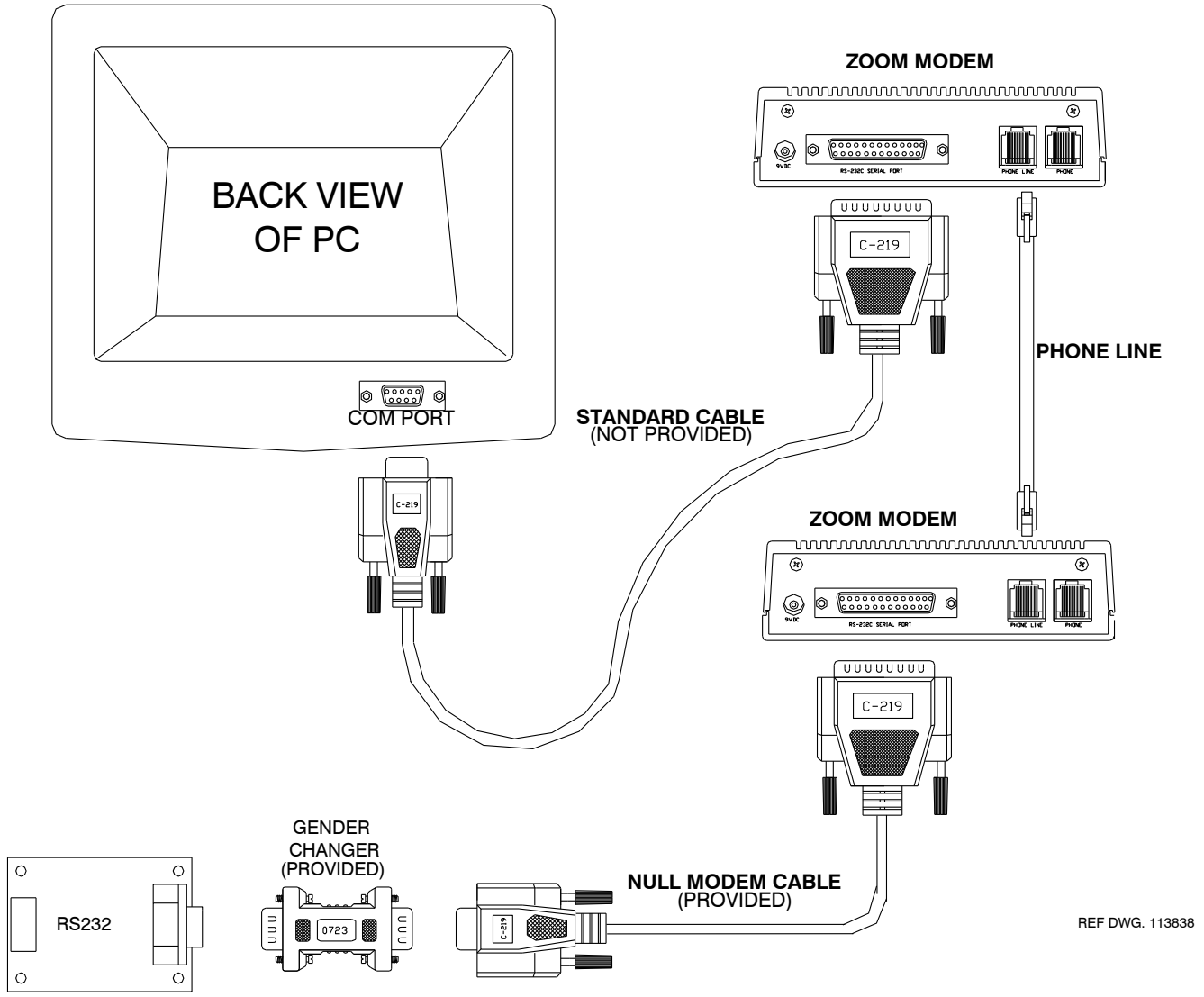
Type the following to make the call:

ATD9, 16105555224 <enter>
ATD is the command
9,16105555224 is the phone number –9, for outside line.

Wait for the message “connected”
To hang up:

Type three plus signs (+++) and wait for the message “OK”
Type ATH0 <enter> to hang up or,
Type ATO0 <enter> to enter online mode again

DETAILED WIRING DIAGRAM



ZOOM MODEM CONNECTION BLOCK DIAGRAM