

Communication Protocol 3003 - Deluxe

Paul Dickerson

Original Issue: 7/25/03

Revision History:

Revision	Date	By	Description
A	7/25/2003	Paul Dickerson	Original issue, based on Protocol 3002. 14 day auto Self Test flag added to :I, :L commands; Boost1, Cut2, Boost2 states added to :S command.
B	8/8/2003	Paul Dickerson	Corrected :K control command to be the same as in Protocol 3002; Binary to decimal conversions indicated in :D, :M, :T and :V query commands.
C	8/26/2003	Paul Dickerson	:S tap state response was 3 = Cut2 and 4 = Boost2; Communication Protocol data type was binary; Added description to :H function; added "loads" to :R response data format; Added "Unused bytes contain an ASCII X" to Note 3; :S comment was "b will be 00 (hex) if not implemented".
D	1/25/2012	Joe Dunne	Corrected checksum note 1. Used to say 2's complement which was wrong.
E	4/23/2014	Joe Dunne	Setup example commands in hex and ascii formats.

A. Controls:

Command (4-7 bytes) :opcode data#<	Response (2-3 bytes) opcode data<	Function	Data Format	Data Type	Data Location	Comments
:A#<	A<	Activate a Self-Test (10 sec)				
:C#<	C<	Cancel Inverter Shutdown				
:G#<	G<	Shutdown Inverter after Pre-Delay				
:Hhhh#<	H<	Shutdown after Pre-Delay (setup by :N) and Wakeup After Delay of hhh	hhh = 0 - 2 ²⁴ seconds	binary	RAM	
:li#<	I<	Write NVR status flags: Initialize Auto-Restart modes & Enable 14 day Self Test	i = bitmap (0/1 = off/on) of: i ₀ - i ₄ = Auto-restart after Shutdown, Delayed Wakeup, LVC, Overload, Overtemp; i ₅ = Enable 14 day auto Self Test	binary	RAM	See the :L query to read NVR status flags
:Jjj#<	J<	Write Unit ID Number	jj = 0 - 2 ¹⁶	binary	RAM	
:Krs#<	K<	Control Relay after Pre-Delay (relay number, relay status)	r = 0-9 = relay number (0 = master, 1-8 = loads); s = 0/1 (off/on) = status of load controlled by relay r	ASCII encoded hex	RAM	
:Nnn#<	N<	Set Pre-Delay Timer	nn = 0 - 2 ¹⁶ seconds	binary	RAM	
:Q#<	Q<	Perform 10 sec Reboot				
:Ww#<	Ww<	Watch Dog Function	w = Disable/Read/Set Watch Dog Delay: w = 0 disables Watch Dog function; w = 1 returns watchdog delay in seconds (if 0, watchdog is disabled); 1 < w < 256 sets watch dog delay to w seconds	binary	RAM	
:Z#<	Z<	Reset Vmin & Vmax register values			RAM	

B. Queries:

Command (4 bytes) :opcode#<	Response (4-8 bytes) opcode data<	Function	Data Format	Data Type	Data Location	Comments
:0x00#<	[0x00]ppXXXX<	Communication Protocol Identification	pp = [0x30][0x03]	BCD	ROM	Command code is hex
:D#<	Daadd<	Voltage Data (VAC, VDC)	aa = 0 - 2 ⁸ : VAC = 1.000*DEC(aa) Volts (120V models) VAC = 1.917*DEC(aa) Volts (230V models); dd = 0 - 2 ⁸ : VDC = 0.1*DEC(dd) Volts	ASCII encoded hex	RAM	
:F#<	Fffffrr<	Main uC Firmware (P/N, Rev)	ffff = Part Number; rr = Revision Letter	ASCII	ROM	
:L#<	Llzzix<	Read: % Load level; Batt Chg Current level; NVR status flags	ll = 0 - 2 ⁸ (%); zz = 0 - 2 ⁸ ; i = bitmap (0/1 = off/on) of: i ₀ - i ₄ = Auto-restart after Shutdown, Delayed Wakeup, LVC, Overload, Overtemp; i ₅ = Enable 14 day auto Self Test	ll,zz = ASCII encoded hex; i = binary	RAM	See the :I Control to write NVR status flags
:M#<	Miimm<	Min, Max Vin (read only)	ii = 0 - 2 ⁸ : VAC = 1.000*DEC(ii) Volts (120V models) mm = 0 - 2 ⁸ : VAC = 1.917*DEC(mm) Volts (230V models)	ASCII encoded hex	RAM	
:P#<	PpppppX<	Power Rating	ppppp = VA rating	ASCII	ROM	
:R#<	Rmr<	Relay Status (master, loads)	m = 0/1 (master off/on); r ₀₋₇ = (0/1 = loads off/on)	binary	RAM	Load relays bit mapped
:S#<	Slcfibp<	Status: low batt; self-test; fault; state flags; % Batt cap; transformer tap	l = 0/1 (battery low/OK); c = 0-5 (OK, Batt Fail, In Progress, Overcurrent, Unknown, Batt Fail & Overcurrent); f = 0-3 (No fault, EPO, OTP, Both); i = bitmap (0/1 = off/on) of State Flags: i ₀ - i ₃ = Inverter, SelfTest, Standby, Idle; b = Percent Battery Capacity; p = 0-4 (Norm, Cut1, Boost1, Boost2, Cut2)	l,c,f,p = ASCII i,b = binary	RAM	b will be [0xFF] if not implemented
:T#<	Ttffffm<	Temperature; Frequency; Freq. Mode	tt = 0 - 2 ⁸ of ADC count; fff = 0 - 2 ¹² : frequency = 0.1*DEC(fff) Hz; m = 0/1 (50/60Hz)	ASCII encoded hex	RAM	
:U#<	Uuu<	Unit ID Number	uu = 0 - 2 ¹⁶	binary	ROM	
:V#<	VaddbXX<	Model Version: nominal Vac; nominal Vdc; # sw load banks	a = 0-4 (100V, 120V, 230V, 208V, 240V); dd = 0 - 2 ⁸ : VDC = 6*DEC(dd) Volts; b = 0-8 (# of switchable load banks)	ASCII encoded hex	ROM	

C. Setup Queries:

Command (4 bytes) :opcode#<	Response (8 bytes) opcode data<	Function	Data Format	Data Type	Data Location	Comments
:0#<	0ffffrr<	USB uC Firmware (P/N, Rev)	ffff = Part Number, rr = Revision Letter	ASCII	USB ROM	
:1#<	1xyyyzz<	USB ID (Mfg, Product, BCD Dev Rel)	xx = [0x09AE]; yy = [0x0001]; zz = [0x0001]	binary	NVR	
:2#<	2aaaaaa<	USB ID (Product Name - 1)	Product name characters 1 thru 6	ASCII	NVR	
:3#<	3aaaaaa<	USB ID (Product Name - 2)	Product name characters 7 thru 12	ASCII	NVR	
:4#<	4aaaaaa<	USB ID (Product Name - 3)	Product name characters 13 thru 18	ASCII	NVR	
:5#<	5aaaaaa<	USB ID (Product Name - 4)	Product name characters 19 thru 24	ASCII	NVR	
:6#<	6aaaa##<	USB ID (Product Name - 5)	Product name characters 25 thru 28; and 16-bit checksum of characters 1-28	ASCII	NVR	

E. Notes:

1. # = Single byte one's complement checksum of all characters (except : and <).
2. < = carriage return.
3. All characters (except checksums) are ASCII encoded unless otherwise indicated. Data characters are shown in lower case.
If indicated, the decoded ASCII character will be interpreted as a hex value. Unused bytes contain an ASCII X.
4. If a command is in process, new commands will be ignored (no response) until the in process command is finished.
5. The :1 response represents: xx = USB Vendor ID for Tripp Lite, which is 09AE; yy = USB Product ID number, which is 0001;
zz = USB BCD Device Release Number, which is 0001.
6. The % Load response to the :L command will be 100% at line rated output current in both Line and Invert modes.
No correction factor is needed in PowerAlert for Invert mode.
7. The product name characters (:1 thru :6 queries) are to be stored in the NVR.
The production test will load in the correct characters for a given model.
Characters 29-30 are a two byte two's complement checksum of characters 1-28.

Hex Example	Ascii example
3A 41 BE 0D	:A¾
3A 43 BC 0D	:C¼
3A 47 B8 0D	:G,
3A 48 00 00 0A AD 0D	{Contains NULL}
3A 49 00 B6 0D	
	{Contains NULL}
3A 4A 00 00 B5 0D	{Contains NULL}
3A 4B 30 30 54 0D	:K00T
3A 4E 00 05 AC 0D	{Contains NULL}
3A 51 AE 0D	:Q®
3A 57 A8 0D	
	:W~
3A 5A A5 0D	:Z¥

3A 00 FF 0D {Contains NULL}

3A 44 BB 0D

:D»

3A 46 B9 0D :F¹

3A 4C B3 0D

:L³

3A 4D B2 0D :M²

3A 50 AF 0D :P⁻

3A 52 AD 0D :R-

3A 53 AC 0D

:S¬

3A 54 AB 0D :T«

3A 55 AA 0D :Uª

3A 56 A9 0D :V©

3A 30 CF 0D	:0İ
3A 31 CE 0D	:1Î
3A 32 CD 0D	:2Í
3A 33 CC 0D	:3İ
3A 34 CB 0D	:4Ë
3A 35 CA 0D	:5Ë
3A 36 C9 0D	:6É