1696/1697/1698 Power Supply

RS232 Remote Communication

Getting Started

1. Make sure you have a STRAIGHT through serial cable. Meaning, it must be pin to pin (pin1 to pin1, pin2 to pin2, ..etc..)
2. Use default windows setting. That is, set Baudrate: 9600 , Data bits: 8, Parity: none, stop bits: 1 (9600-8-N-1)
3. If you are using hyperterminal, make sure you setup "ASCII setup", which can be found under File->Properties->Settings tab->Click ASCII setup button. In the setup, DO NOT append line feeds under "ASCII Sending", so keep it unchecked. You can check the box for "Echo Typed Characters locally". In "ASCII Receiving", you can append line feeds. (Hyperterminal is good for testing commands to ensure you have correct setup and connection)

Command Sets

Notes:

- All the examples below are shown with <address> = 00
- All commands are case sensitive and must be in capital letters.
- There are no spaces whatsoever in any command strings. This means no spaces before, in between, or after any parts of the strings.
- <address> = 00 to 99 - Address is used only when multiple power supplies are used/connected simultaneously over RS-485 configuration. Each supply needs to have a unique address so that subsequent commands will control the power supply with the assigned address. In RS-232 configuration, this setting is ignored, so users can use any values within the range so as long as it is consistent across all other subsequent commands.
- <CR> = Carriage return. This is the terminal character used to terminate each string. It is, in some programming language, the equivalent of "\r", or hex representation of 0xD and decimal representation of 13.
- Anything in **BLUE** colors indicate the command string to send to the instrument.
- Anything in **RED** colors indicate the return string that returns immediately after the associated command string is sent.
- Anything in [] indicates the returned character or characters.
  [OK] means the characters "OK" is returned.
  [CR] means the carriage return or "\r" character is returned.
  [???] means some numeric value of three digits is returned.
  [??] means some numeric value of two digits is returned.
- XXX indicates the maximum limit of the power supply, either for voltage or for current as indicated.
- All voltage values are in the format of XX.X V and all current values are in the format of X.XX A. The decimal place is fixed.
- Anything in <> indicates a user defined parameter that is part of the command string.

Command Protocol	Description	Example (address = 00)
SESS<address><CR> **[OK][CR]**	Disables front panel keypad and sets power supply into remote mode	**SESS00<CR>**
ENDS<address><CR> **[OK][CR]**	Enables front panel keypad and sets power supply into local mode	**ENDS00<CR>**
Setting up communication		
CCOM<address><RS><RS485 address><CR> **[OK][CR]** **Note:** <RS> = 0 for RS232 = 1 for RS485 <RS485 address> = 000 to 031	Changes the power supply between RS232 and RS485 communication. When this command is sent and RS485 address is changed, commands to change the power supply in RS-485 configuration will also need to change to the new address. For example, suppose we set RS 485 address to 10. The <address> parameter for all other commands on the same power supply will need to be changed to "10" (i.e. SESS10, GCOM10, etc.)	**CCOM001002<CR>** **Note: This will set supply into RS485 mode with address set to 002.**
GCOM<address<CR> **[OK][CR]**	Gets the RS485 address configured on the power supply.	**GCOM00<CR>**
Setting up power supply parameters		
VOLT<address><voltage><CR> **[OK][CR]** **Note:** <voltage> = 010 to XXX where XXX is the maximum voltage rated by the power supply. Format: XX.X V i.e. 234 = 23.4 V Minimum value: 010 -> 1.0 V (decimal point does not change)	Sets the voltage output of the power supply.	**VOLT00123<CR>** **Note: This will set voltage to 12.3V**
CURR<address><current><CR> **[OK][CR]**	Sets the current limit of the power supply	**CURR00456<CR>** **Note: This will set current limit to 4.56A**

Note: <current> = 001 – XXX where XXX is the maximum current rated by the power supply Format: X.XX A i.e. 123 = 1.23 A (decimal point does not change)		
SOVP<address><voltage><CR> **[OK][CR]** **Note:** <voltage> = 010 to XXX where XXX is the maximum voltage rated by the power supply. Format: XX.X V i.e. 234 = 23.4 V Minimum value: 010 -> 1.0 V (decimal point does not change)	Sets the upper voltage limit of the power supply (OVP limit)	**SOVP00105<CR>** **Note: This will set the upper voltage limit (OVP) to 10.5V**
GETS<address><CR> **Voltage[???]Current[???][CR]** **[OK][CR]**	Gets the set voltage and set current values from power supply.	**GETS00<CR>** **i.e. If set voltage = 12.3 V and set current = 4.56 A, the return string will be:** **123456[CR]** **[OK][CR]**
GOVP<address><CR> **Voltage[???][CR]** **[OK][CR]**	Gets the upper voltage limit setting (OVP)	**GOVP00<CR>** **i.e. If upper voltage limit = 10.0 V, the return string will be:** **100[CR]** **[OK][CR]**
GETD<address><CR> **Voltage[???]Current[???][mode][CR]** **[OK][CR]** **Note:** [mode] = 0 for CV mode = 1 for CC mode	Gets the measured voltage and current reading and the mode of operation.	**GETD00<CR>** **i.e. If measured/read voltage = 1.0 V and current = 4.56 A and in CC mode, return string will be:** **0104561[CR]** **[OK][CR]**
GMAX<address><CR> **Voltage[???]Current[???][CR]** **[OK][CR]**	Gets the maximum voltage and current settable by the power supply	**GMAX00<CR>** **i.e. 1696 rated for 20.0 V and 9.99 , return string will be:** **200999[CR]** **[OK][CR]**
Output control		

SOUT<address><status><CR> **Voltage[???]Current[???][CR]** **[OK][CR]** **Note:** <status> = 0 for Enable output (ON) = 1 for Disable output (OFF)	Enables or disables the output of the power supply.	**SOUT000<CR>** **Note: This will enable the output (ON).**
Display status		
GPAL<address> CR> **[68 characters][CR]** **[OK][CR]** **Note:** [68 characters] – The 68 characters represent all the LCD display information. (See "Explain of GPAL Command" section for details.)	This gets all the information as displayed on the LCD screen. This command is useful to get the current status of the power supply, including measured voltage, current, and power, set voltage and current, key lock status, remote status, output status, and more.	**GPAL00<CR>**
Memory Presets		
PROM<address><memory_location><voltage><current><CR> **[OK][CR]** **Note:** <memory_location> = 1 to 9 ; <voltage> = voltage value to set for the assigned memory location. Format: XX.X V <current> = current value to set for the assigned memory location. Format: X.XX A (decimal point does not change)	Saves the defined voltage and current into the specified memory location.	**PROM005145020<CR>** **Note: This will set 14.5 V and 0.20 A into memory location 5.**
GETM<address><CR> **Memory 1 Voltage[???]Current[???][CR]** **Memory 2 Voltage[???]Current[???][CR]** **….. ……** **……..** **Memory 9 Voltage[???]Current[???][CR]** **[OK][CR]** **Note:** Only the characters in [] are returned. The words "Memory 1" or "Voltage" or "Current"	Returns all of the voltage and current values that are stored in all memory locations of the power supply	**GETM00<CR>** **Note: This will return all voltage and current values stored into memory of the power supply.** **i.e. If location 1 has 1.0 V/1.00A, location 2 has 2.0V/2.00A…..location 9 has 9.0V/9.00A, the return string will be:**

are NOT part of the return string.		**010100[CR]** **020200[CR]** **…….** **090900[CR]**
GETM<address><location><CR> **Voltage[???]Current[???][CR]** **[OK][CR]** **Note:** <location> = 1 to 9 Only the characters in [] are returned. The words "Voltage" or "Current" are NOT part of the return string.	Returns only the stored voltage and current settings from the specified memory location.	**GETM002<CR>** **Note: This will return the voltage and current values stored into memory location 2.**
RUNM<address><location><CR> **[OK][CR]** **Note:** <location> = 1 to 9	Recalls the voltage and current values stored in the specified memory location	**RUNM006<CR>** **Note: This will recall the voltage and current settings stored in memory location 6.**
Timer Program		
PROP<address><location><voltage><current><minute><second>CR> **[OK][CR]** **Note:** <location> = 00 to 19 This is the location for the step. Timed program can store up to 20 steps (i.e. 00 – 19) <voltage> = 010 to XXX Format: XX.X V <current> = 001 to XXX Format: X.XX A <minute> = 00 to 99 This is the time in minutes to hold the programmed step values <second> = 00 to 59 This is the time in seconds to hold the programmed step values	Sets the parameters (voltage, current, minutes, seconds) for one step indicated by the step location.	**PROP00151234560435<CR>** **Note: This will set the parameters:** **Voltage = 12.3 V,** **Current = 4.56 A,** **Minutes = 4** **Seconds = 35** **Step = 15**
GETP<address><CR> **Program 00** **Voltage[???]Current[???]Minute[??]Second[??][CR]** **Program 01Voltage[???]Current[???]Minute[??]Second[**	Returns all 20 steps of the timed program stored parameter values from memory.	**GETP00<CR>**

??][CR] **….. ……** **…….** **Program 19** **Voltage[???]Current[???]Minute[??]Second[??][CR]** **[OK][CR]** **Note:** Only the characters in [] are returned. The words "Program 00","Voltage", "Current", "Minute", or "Second" are NOT part of the return string.		
GETP<address><location><CR> **Voltage[???]Current[???]Minute[??]Second[??][CR]** **[OK][CR]** **Note:** <location> = 00 to 19 Only the characters in [] are returned. The words "Voltage", "Current", "Minute", or "Second" are NOT part of the return string.	Returns only the specified step location's stored parameters in timed program.	**GETP0010<CR>** **Note: This will return the voltage, current, minutes, and seconds parameters stored in step location 10 of the timed program.**
RUNP<address><cycle>CR> **[OK][CR]** **Note:** <cycle> = 0000 to 0256 This defines how many cycles to repeat the timed program, up to 256 cycles 0000 = infinite cycles	Runs the timed program and repeats for the specified number of cycles.	**RUNP000182<CR>** **Note: This will run the timed program and repeat for 182 cycles.**
STOP<address>CR> **[OK][CR]**	Stops the timed program from running.	**STOP00<CR>**

Explanation of GPAL command:

As mentioned previously, the GPAL command obtains all LCD display information. It is heavily encoded in hexadecimal representations that require conversion to different bits of binary, as well as some conversion for 7-segment BCD.

For a clear description, we will emphasize the following:

- Partitioning string of information
- Explanation of representation of partitions
- Converting partitions into binary
- Converting binary to decimal based on 7-segment BCD decoding

Partitioning string of information

When GPAL command is sent to the power supply, the supply returns a string of information containing a total of 68 characters. These characters represent all the LCD display information on the power supply. The string must first be partitioned so that the information can be analyzed in an organized fashion.

Here is an example of a return string after sending GPAL command:

00>=4?3?0866=6?4?0??66665;000000000111100>=4?010=;3?3?11000110101011

Here's a chart of each character with the numbering below for easier reference. For clarity, it is split into two rows of 34. If you count, it totals 68 characters.

00>=4?3?0866=6?4?0??66665;000000000111100>=4?010=;3?3?11000110101011

0	0	>	=	4	?	3	?	0	8	6	6	=	6	?	4	?	0	?	?	6	6	6	6	5	;	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

0	1	1	1	1	0	0	>	=	4	?	0	1	0	=	;	3	?	3	?	1	1	0	0	0	1	1	0	1	0	1	0	1	1
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68

Here are the characters' representation description

Characters	Value from above example	Description
1-8	**00>=4?3?**	**Reading Voltage in Volts**
9	**0**	**Not used**
10-17	**866=6?4?**	**Reading Current in Ampere**
18	**0**	**Not used**
19-26	**??66665;**	**Reading Power in Watts**
27	**0**	**Not used**
28-31	**0000**	**Minutes on Timer**
32-35	**0000**	**Seconds on Timer**
36	**1**	**"Timer" Display**
37	**1**	**":" Display for timer function**
38	**1**	**"m" indicator Display. Not used**
39	**1**	**"s" indicator Display. Not used**
40-45	**00>=4?**	**Setting Voltage**
46	**0**	**"V-const" indicator Display**
47	**1**	**"V-set" indicator Display**
48	**0**	**"V" Display**
49-54	**=;3?3?**	**Setting Current**
55	**1**	**"I-const" indicator Display**
56	**1**	**"I-set" indicator Display**
57	**0**	**"A" Display**
58-59	**00**	**Program number**
60	**1**	**"Program" Text Display**
61	**1**	**P-bar. Not used.**
62	**0**	**"Setting" Text Display**
63	**1**	**Key Lock icon Display**
64	**0**	**Key Unlock icon Display**
65	**1**	**"Fault" indicator Display**
66	**0**	**Output ON icon Display**
67	**1**	**Output OFF icon Display**
68	**1**	**Remote mode icon Display**

Explanation of representation of the partitions

Characters:	Explanation
1-8:	Reading Voltage – These 8 characters represent the reading voltage value from the display in volts. This is the voltmeter reading from the supply that indicates the voltage at the terminal output. *(see next section for decoding description)*
10-17:	Reading Current – These 8 characters represent the reading current value from the display in amps. This is the ammeter reading from the supply that indicates the current at the terminal output. *(see next section for decoding description)*
19-26:	Reading Power – These 8 characters represent the reading voltage value from the display in watts. This is the power reading from the supply that indicates the power at the terminal output. *(see next section for decoding description)*
28-31:	Minutes on Timer – This is the minutes set when using Timer function. *(see next section for decoding description)*
32-35:	Seconds on Timer – This is the seconds set when using Timer function. *(see next section for decoding description)*
36:	Timer Display – This indicates the "Timer" text displayed on the screen. 1 means not displayed. 0 means displayed.
37:	Colon Display – This indicates the ":" displayed on the screen when Timer function is enabled or being setup. 1 means not displayed. 0 means displayed.
40-45:	Setting Voltage – This indicates in volts the voltage that the supply is setup for. *(see next section for decoding description)*
46:	"V-const" Display – This indicates the "V-const" indicator display, which also indicates the supply in CV mode when it is displayed. 0 means ON. 1 means OFF (not in CV).
47:	"V-set" Display – This indicates the "V-set" indicator display, which also indicates the voltage set mode. When displayed, voltage can be set. 0 means ON. 1 means OFF (not displayed).
48:	"V" Display – This will always be on to indicate the unit on the display for voltage values. 0 means ON. 1 means OFF.
49-54:	Setting Current – This indicates in amps the current that the supply is setup for. *(see next section for decoding description)*
55:	"I-const" Display – This indicates the "I-const" indicator display, which also indicates the supply in CC mode when it is displayed. 0 means ON. 1 means OFF (not in CC).
56:	"I-set" Display – This indicates the "I-set" indicator display, which also indicates the current set mode. When displayed, current can be set. 0 means ON. 1 means OFF (not displayed).

57: "A" Display – This will always be on to indicate the unit on the display for current values. 0 means ON. 1 means OFF.

58-59: Program Number – This indicates the program number shown on display when configuring programmed values for storage. *(see next section for decoding description)*

60: "Program" Text Display – This indicates the text "Program" is displayed. This should only be on when setting up programmed values into the power supply. 1 means OFF (not displayed). 0 means ON.

62: "Setting" Text Display – This will always be on to indicate which group of values for voltage and current is for displaying the set voltage and current. 0 means ON. 1 means OFF.

63: Key Lock Icon Display – This indicates the display of the key lock icon. When displayed, the instrument front keys are locked and cannot be accessed locally. 0 means ON. 1 means OFF. When this is 0, character 64 will be 1. When this is 1, character 64 will be 0.

64: Key Unlock Icon Display – This indicates the display of the key unlock icon. When displayed, the instrument front keys are unlocked and local control is allowed. 0 means ON. 1 means OFF. When this is 0, character 63 will be 1. When this is 1, character 63 will be 0.

65: "Fault" Indicator Display – This is the "fault" indicator display, which only displays when OVP (over voltage protection) is tripped. Under normal usage, this should be off. 0 means ON, 1 means OFF.

66: Output ON Icon Display – This refers to the icon display for output ON. When this is indicated (ON), output terminals are ON. If not, then output is off. 0 means ON for display. 1 means OFF. If this is 0 (ON, output is on), than character 67 will be 1 (OFF, output off icon disabled).

67: Output OFF Icon Display – This refers to the icon display for output OFF. When this is indicated (ON), output terminals are OFF. If not, then output is on. 0 means ON (which means output is disabled). 1 means OFF (this indicator is not displayed). If this is 0 (ON, or output is OFF), than character 66 will be 1 (OFF, output is not ON).

68: Remote Icon Display – This indicates the icon display for remote connection. When supply is in remote mode, this display will be ON. 0 means ON. 1 means OFF.

Converting partitions into binary

Some parts of the string require decoding to binary values before they can be converted into a more readable format. The following characters will require this conversion process:

1-8 (Reading Voltage)
10-17 (Reading Current)
19-26 (Reading Power)
28-31 (Timer minutes)
32-35 (Timer seconds)
40-45 (Setting Voltage)
49-54 (Setting Current)
58-59 (Program number)

The conversion of any of the above listed characters requires changing from decimal to a 4-bit binary value. As noticed already, some characters have symbolic representations such as ; or ?. These symbols represent values in hexadecimal, and including the numbers 0-9 they are actually character representations of decimals 48-63. In hex, this is equivalent to 0x30-0x3F. We are only interested in looking at the least significant digit in this group of hex numbers, meaning we are looking at 0x00-0x0F.

Each of the characters from the above list will need to be converted to a 4-bit binary format. Below is the conversion table.

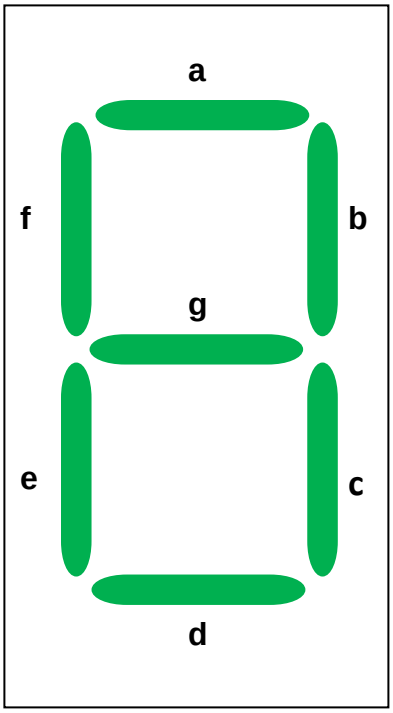
Character	4-bit Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
:	1010
;	1011
<	1100
=	1101
>	1110
?	1111

Once these values are converted, they must be decoded based on the 7-segment BCD decoding of the display. Follow the next section for details.

Converting binary to decimal based on 7-segment BCD decoding

Every group of characters as listed from previous section requires this conversion process from binary to decimal based on 7-segment BCD decoding of the display. This is necessary because the GPAL command grabs the low-level bits data directly from the LCD display. For better understanding, please see below a summary of 7-segment BCD display and what they represent.

7-segment BCD display



Digit display	Segments (1= ON, 0 = OFF)						
	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0
2	1	0	1	1	0	1	1
3	1	0	0	1	1	1	1
4	1	1	0	0	1	1	0
5	1	1	0	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	0	0	1	1	1
8	1	1	1	1	1	1	1

Converting binary to decimal

The table above for the 7-segment display is listed in the order g-a instead of a-g because in the conversion process, the bit order to use will be from g-a. When a group of characters (i.e. 1-8 for reading voltage) are converted into 4-bit binary, they will be divided into groups of 8 bits. For each group of 8 bits, the first bit represents whether a decimal will follow or not. If it is a 0, no decimal will be placed after the digit. If it is 1, a decimal is placed after the digit. The remaining 7 bits will represent the 7 bits as indicated in the above table for the 7-segment display. The digit representation of those bits is as indicated in the table.

Here is an example taken from the sample string in previous section:

Example: Characters 1-8: Reading Voltage

00>=4?3?

Step 1: First we convert the string into 4-bit binary values as follows:

0000 0000 1110 1101 0100 1111 0011 1111

Step 2: Divide them into groups of 8 bits as follows:

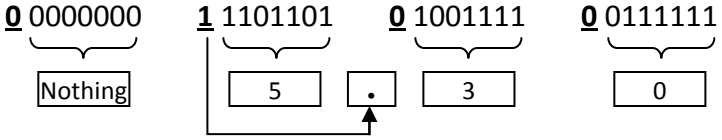
00000000 11101101 01001111 00111111

Step 3: Check for the first digit of each group to see if decimal will be placed after the digit.
0 = no decimal ; 1 = decimal will be placed

0 0000000 **1** 1101101 **0** 1001111 **0** 0111111

Indicates decimal will be placed after digit

Step 4: Now take the remaining 7 bits of each divided group of bits and use the 7-segment decoding chart above to convert values into the digits they represent. The decimal from previous step is placed after the digit converted from within its original group of 8 bits.



Step 5: Now put the converted values together and it reads:

5.30 (in Volts)