

# Versalent

Mar 2018

---

## PS2PROX PS2 Keyboard Serial Converter Manual

Version 1.00  
Revised Mar 16, 2018



### General Description

**NOTE:**

This manual describes the PS2PROX which is upgraded version of the PS2PRO. The basic difference between the two types is that the PROX models allow the user to program up to 93 keyboard keys to output user-defined strings of 0-10 characters, rather than the maximum of 16 strings of 6 characters of the PS2PRO.

A Windows® based configuration program is provided with PS2PROX .. 'PS2PROX Config' to allow the definition of custom output strings. This program allows you to choose a string format (from 1 characters long to 10 characters long) then simply select a key from the keyboard, and enter the custom output character or string which the PS2PROX will generate when that key is typed. See the table below. Once sent to the PS2PROX you can read the existing 'key mappings' back by clicking a single button, and/or modify or remove these mappings.

<b>Output String Length</b>	<b>Maximum Number of Override Keys</b>
1	93
2	85
3	63
4	51
5	42
6	36
7	31
8	28
9	25
10	23

This table shows the maximum number of keys which can be overridden based on the length of the output string chosen. This flexibility in trading string length for number of override keys allows the user to easily adapt the PS2PROX to their particular application.

PS2PROX is a small module (2.6" x 1.7" x 0.8") that translates PS2 PC keyboard output to the standard RS-232 characters printed on the keys so non-PC based hosts can easily use these low cost keyboards. The PS2 keyboard protocol has real-time requirements which can interfere with normal operation if not off-loaded to an auxiliary processor. And the keyboard's keycodes can be complex sequences of multiple codes. PS2PROX simplifies all that and provides a simple, standard RS-232 output which can interface to almost any computer system – even small micros which do not have the resources to manage the real-time PS2 protocol and still do other work.

PS2PROX connects to a keyboard through the standard 6-pin PS2 keyboard connector, and to your host system through a standard DB9 (Female)\* connector. It analyzes the keyboard's scan set 2 keycodes (the keyboard's default scan code set) and outputs the ASCII character printed on the key. **(PS2PROX can be programmed to output 0-10 character strings instead)** . It also translates all the non-ASCII keys and generates 8-bit serial characters for those as well so that your host can recognize and act on all keys including function keys, arrow keys, keypad keys with Numlock/Ctrl/Alt both on and off etc.

Keyboard power is provided through the PS2 connector, and therefore through PS2PROX. Various power options allow you to supply power for both the keyboard and PS2PROX via the DB9 connector, or with the optional 'power brick' wall supply. And there are several models which provide DTE or DCE connection options so you won't need a null-modem adapter to adjust connector signal/pin positions.

Keyboard LEDs are handled transparently so if the user presses CAPS LOCK, the LED is turned on automatically and characters arriving from PS2PROX are uppercased so no host action is required. When the PS2PROX is powered and operating a small **green** LED flashes every ½ second as a visual indication that power is applied and it is operating.

On power-up, the PS2PROX waits up to 1.5 seconds for the keyboard to send its self-test result code. (The keyboard sends 0xAA to indicate PASS, or 0xFC to indicate FAIL.) The PS2PROX then sends one serial character (0xE1) to indicate that the keyboard is ready, or 0xE2 to indicate that the keyboard either failed its self-test .. or did not respond at all.

## **Reliability Features:**

PS2PROX implements an internal watchdog timer, and an internal brown-out detector. Should its microcontroller get disrupted through static discharge or other temporary interference, the watchdog will automatically reset the unit so that normal operation resumes with no user intervention. Or should power droop below an operational threshold the brown-out detector will suspend operation until power is restored to a normal level. At that point it will resume operation from a reset state again with no user intervention.

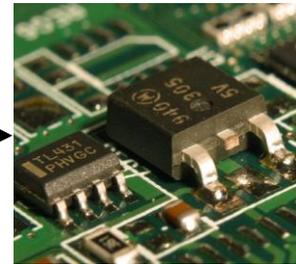
\*Note that the connector is female even for the optional DTE pin configurations which are more commonly a male – like those on a (DTE) PC COM port. If male is needed, a small ‘gender-bender’ adapter is recommended.



PS/2 Keyboard



PS2PROX



Your product w/RS-232 Serial Port

### Typical Application

## Baud Rate/Parity Control:

Baud and parity are set using the 5 internal shorting jumpers as shown below. To open the snap-together plastic case and access them, there are 2 ‘screw-driver slots’ on each side of the case at the seam. Gently pry the case open using a small flat-blade screwdriver at any one of these slots. After the first internal latch releases, the others release even easier and the two halves of the case separate.

PS2PROX can be set to any of the baud rates/parity settings shown in TABLE 1. To configure it the plastic case is opened, and 5 internal shunt blocks are moved to the positions shown. These settings are also noted on the inside of the case cover. Note that the 3 shunt blocks on the left control the baud rate and the right two select parity. The table shows shaded blocks where shunts are to be installed to select the specified settings. After changing these settings the PS2PROX must be powered off/on to reset and accept the new serial settings.

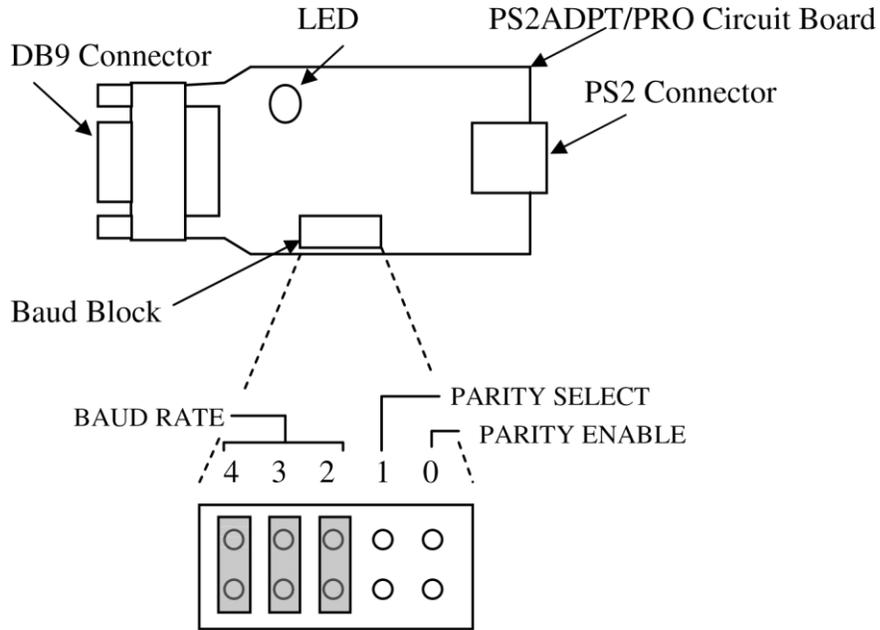
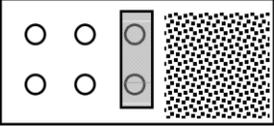
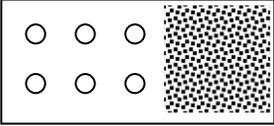
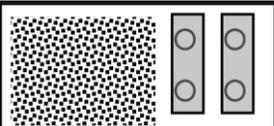
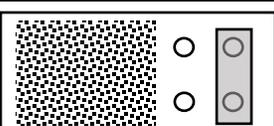
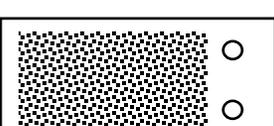
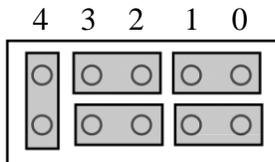


DIAGRAM	BAUD RATE	PARITY
	1200 bps (110 bps w/LB option)	See below
	2400 bps (300 bps w/LB option)	See below
	4800 bps (600 bps w/LB option)	See below
	9600 bps (1200 bps w/LB option)	See below
	19.2k bps (2400 bps w/LB option)	See below
	38.4k bps (4800 bps w/LB option)	See below

	57.6k bps (9600 bps w/LB option)	See below
	115k bps (19.2k bps w/LB option)	See below
	See above	Parity Enabled, ODD Parity
	See above	Parity Enabled, EVEN Parity
	See above	Parity Disabled

**TABLE 1**

Note that any shunt blocks in a horizontal orientation have no effect on serial settings and are typically used for storing the shunts. PS2PROX devices are shipped in the following default configuration .. 9600 baud, no parity. Low-Baud (LB) option units ship with the default 1200 baud, no parity settings. (All the horizontal blocks have no effect; the only effective one is the vertical one to the far left.)



## **RS-232 Signal Compatibility:**

The PS2PROX is compatible with standard RS-232 signals levels which go both positive and negative (above and below 0 volts) and is also compatible with the non-standard 0-5

volt signal levels used in some systems. This is because the internal Texas Instruments MAX202 RS-232 driver has a receiver threshold that is approximately +1.2 volts above the RS-232 ground. So these ‘single-ended’ signals which do not go below ground provide a sufficient RS-232 signal level for the PS2PROX to operate reliably. Please refer to:

<http://www.ti.com/lit/gpn/max202> for more technical details of the driver's capabilities.

## Power Control:

The standard PS2 keyboard requires +5VDC to operate and so does the PS2PROX. Keyboard power is applied through the PS2PROX's PS2 connector so external power need only be applied to the PS2PROX. There are two ways to do this:

- 1) Apply power to a pin of the DB9 connector
- 2) Plug a ‘power brick’ into the PS2PROX

### Applying Power Through the DB9 Connector:

Power can be applied to either PIN 4, or PIN6 of the DB9 depending on the model. See table below. During assembly different options are installed which provide this flexibility. In addition the voltage applied can also be selected again, based on the model selected. If your system has +5VDC @ 250mA available\*\*, and the cable to the PS2PROX will be fairly short, you can select one of the 5VDC input models (voltage drop through the cable will be minimal). However if the cable will be long and significant voltage could be dropped in the cable, a better option is the 6-12VDC model. The low drop-out regulator in the PS2PROX will supply the keyboard with +5VDC.

In addition to power options there are various models which configure the DB9 connector as DTE or DCE. Choose the model appropriate to mate to your DB9 with no null-modem adapter.

\*\*Note that although PC keyboards are often rated at 250mA maximum, typical keyboards require considerably less current. Many require in the range of 30-60mA. Selection of which voltage option is appropriate is left to the system designer.

### PS2PROX-xxx MODELS AVAILABLE

Model#	Features
PS2PROX-014	DB9 = DCE, +5VDC power applied to DB9 Pin 4
PS2PROX-016	DB9 = DCE, +5VDC power applied to DB9 Pin 6
PS2PROX-024*	DB9 = DCE, 6-12VDC power applied to DB9 Pin 4 or use optional 6V wall supply

PS2PROX-026*	DB9 = DCE, 6-12VDC power applied to DB9 Pin 6 or use optional 6V wall supply
PS2PROX-034	DB9 = DTE, +5VDC power applied to DB9 Pin 4
PS2PROX-036	DB9 = DTE, +5VDC power applied to DB9 Pin 6
PS2PROX-044*	DB9 = DTE, 6-12VDC power applied to DB9 Pin 4 or use optional 6V wall supply
PS2PROX-046*	DB9 = DTE, 6-12VDC power applied to DB9 Pin 6 or use optional 6V wall supply

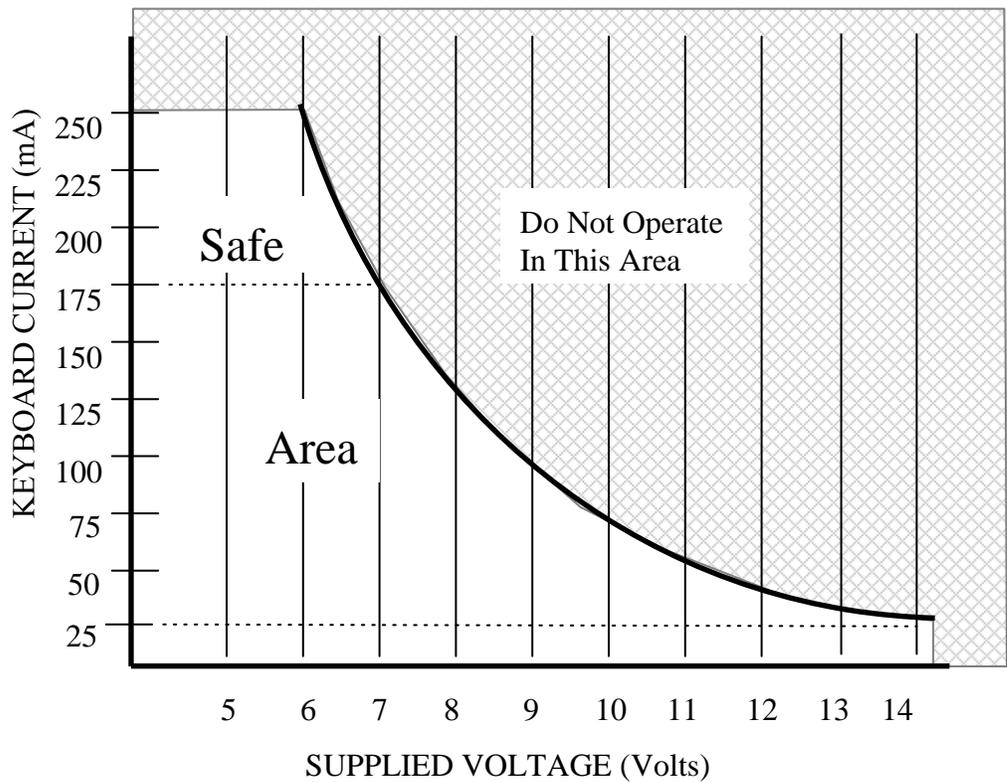
\*These model numbers have a suffix of CN or CP or blank indicating the polarity of the external power supply connector. CN (or blank) = Center Negative, CP=Center Positive.

## **TABLE 2**

*Note that the COMx ports on a standard PC are configured as DTE and therefore its RS-232 signals mate directly to the first four (DCE) models in the table above. That is, the PC's Tx signal connects to the PS2PROX Rx signal, and the PC's Rx signal connects to the PS2PROX Tx signal with a standard straight-thru M/F DB9 cable.*

When applying DC power to the PS2PROX via the DB9 connector, use the FIGURE 1 power dissipation graph to ensure that the PS2PROX is operating in a safe power range for your keyboard current. Note that this is a room temperature graph. See **Environmental Specifications** section to derate input voltage for other temperatures.

## **Room Temperature Safe Operating Area**



**FIGURE 1**

**Applying Power Using the ‘Power Brick’:**

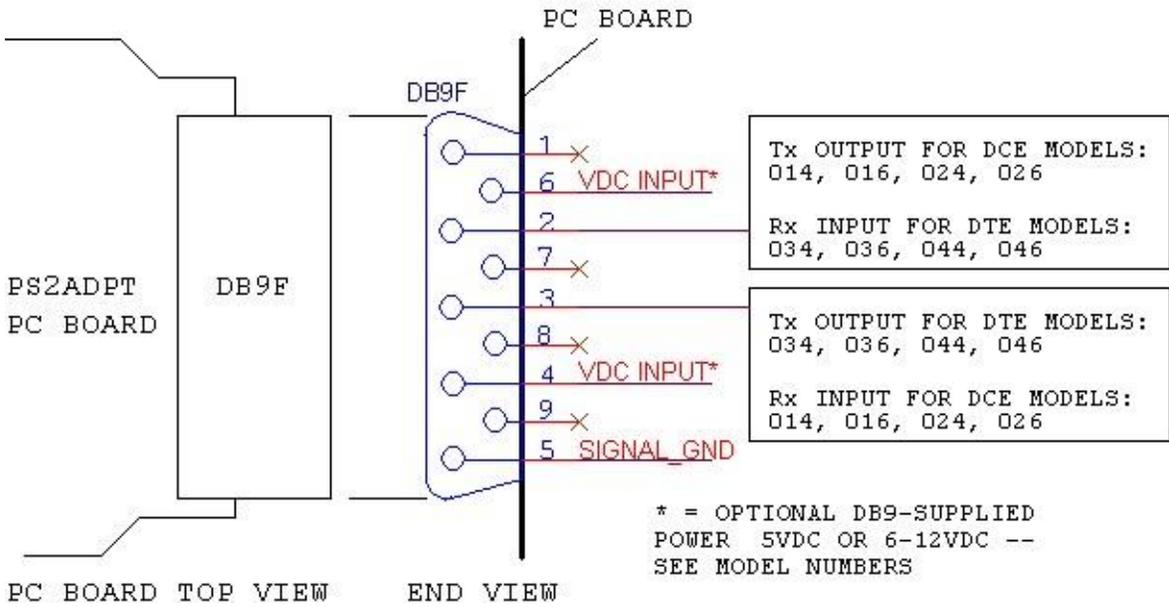
Power can also be applied using the optional ‘6VDC power brick’ which is the optional wall supply. This option is useful when it is not convenient to route power through the DB9 connector, or additional power is not available from the system. Choose an appropriate model that allows operation with the wall supply (Model PS2-PWR-CN/CP) which plugs into any standard 110VAC power outlet. The CN or CP suffix indicates the connector polarity. CN or blank = Center Negative, CP = Center Positive. This must match the suffix of the PS2PROX, i.e. PS2PROX-024CP.

**PS2PROX Physical, Electrical, Environmental Specifications**

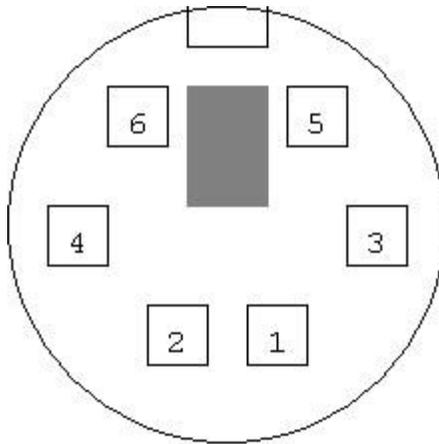
**Physical:**

Size:	2.6” X 1.7” X 0.8”
Weight:	2.6 oz
Keyboard Connector:	Standard PS2 6-pin mini DIN (female)
Host Connector:	Standard DB9 (female)

Case Color:	Black (Ivory available special order)
Power Connector:	5mm X 2.1 mm (male, CN=Ctr Negative, CP=Ctr Positive)



**PS2PROX DB9 (Female) Pin Assignments Diagram**



**PS2PROX PS2 (Female) Connector Pin Assignments Diagram**  
(Mates to standard PS2 6-pin Male Keyboard Connector)

## Electrical:

Absolute Maximum Input Voltage	+15VDC (Internal Regulator Models only)*
Nominal Voltage Input:	+5VDC +/- 5% or 6-12VDC (per model number)
Current Input:	15mA maximum plus keyboard current
Baud Rate:	1200 to 115k selectable via shorting blocks
Parity:	Selectable ODD/EVEN/NONE
DB9 Connector Configuration:	DTE or DCE (per model number)

\*Includes Models -024, -026, -044, -046

The PS2PROX is designed for simplicity – by default it outputs a single ASCII character (the one printed on the keyboard key) when the key is pressed. **Or it can be user programmed to output string of 0-10 characters for a single key press.** If the key is held and the keyboard auto-repeats the key, then the PS2PROX also auto-repeats the serial character (**or string**) output. Although the keyboard itself sends multiple keycodes when keys are pressed, and even when keys are released, the PS2PROX does not. It only sends the serial character/string information you are interested in which is when a key is pressed or when the keyboard auto-repeats a key which is held depressed.

Since the standard 7-bit ASCII character set contains only 128 characters (0x00 thru 0x7f), in order to support the keyboard's function keys, keypad keys and CTRL keys, the PS2PROX sends 8-bit characters to identify these 'non-printable' keys. And to support the use of the ALT key as a character-modifier the PS2PROX sends two characters to identify ALT-x keys. Please refer to <http://www.versalent.biz/manuals/PROXKeyMap.pdf> which is a diagram of a 102-key PS2 keyboard showing all key positions, and the associated character(s) which the PS2PROX generates when the key is pressed. You will notice that there are 5 characters (hex values) on each key to identify what 8-bit characters are output when each key is pressed:

- 1) Normal keypress .. with no other keys pressed
- 2) Key pressed with SHIFT pressed (or CAPSLOCK active)
- 3) Key pressed with ALT pressed
- 4) Key pressed with CTRL pressed
- 5) Key pressed with NUMLOCK active

You will notice that the NUMLOCK, CAPS LOCK, SHIFT, ALT and CTRL keys do not cause any characters to be sent when pressed by themselves. These are 'modifier' keys only – which affect the values that other keys send when pressed.

## Environmental:

Max Operating Temperature:	+85°C Derate by 12°C per 100mW Power dissipation
Min Operating Temperature:	0°C
Max Storage Temperature:	+100°C
Min Storage Temperature:	-40°C
Humidity:	Non-condensing at all temperatures

$$\text{Power Dissipation} = (\text{Input Voltage} - 5) * (\text{keyboard current} + 0.010\text{A})$$

## Derating Example:

$$\text{Keyboard Current} = 65\text{mA}, \text{ Input Voltage} = 10\text{VDC}$$

$$\text{Power Dissipation} = (10 - 5) * (.065 + .01) = 375\text{mw}$$

$$\text{Required derating} = 12^\circ\text{C} * (375\text{mw}/100\text{mw}) = 45^\circ\text{C}$$

$$\text{Max Ambient Operating Temp} = +85^\circ\text{C} - 45^\circ\text{C} = 40^\circ\text{C}$$

## How to Configure PS2PROX for Custom Character/String Output

- 1) Download and install 'PROX Config Windows program
- 2) Connect PS2PROX to a PC COM port. [Note that if you purchased a model that does NOT accept the external AC wall-adapter power (-014,-016,-034,-036), you will need a power-tap cable adapter to allow the application of +5VDC to the PS2PROX DB9 connector while connected to the COM port for programming.] You do NOT need a PS2 keyboard plugged into the PS2PROX during programming and removing it reduces confusion.
- 3) Set COM port to same serial communication settings as PS2PROX
- 4) Start the PROX Config program, and configure it to use the connected COM port at the PS2PROX's baud and parity settings.
- 5) Choose a String Configuration from the drop-down box. Typically you will set the length of the strings to equal the longest string any of your keys will generate. The string length chosen determines the number of available keys which can be 'overridden' with strings.

- 6) To override a key-cap character, place the the mouse in a Keyboard Key box and press the single keyboard key (on your PC Keyboard) for which you want a custom output. Immediately to its right enter the 0-10 character string to be output instead of the default character output. Repeat for as many keys as needed.
- 7) You can SAVE your key overrides in a file which you can later recall using File-Save/SaveAS. Click the 'Send' button, which downloads these key definitions to the PS2PROX and also performs a Verification read-back. Green checkmarks appear next to each string as its successful programming is verified.
- 8) At any time you can verify that the configured key outputs stored in the PS2PROX match those on-screen. So for instance if you have SAVED your set of key configurations to a disk file you can later open that file, and use the 'Verify' button to check that they match. Any strings that do not match will be marked on-screen with a red question mark instead of the green checkmark.

### More Details For Each Step Above:

**Step 1:** The *PS2PROX Config* program is available from the DOWNLOADS section of the <http://www.versalent.biz> website.

**Step 2:** The DCE models (-014, -016, -024, -026) connect directly to a PC COM port with no pin-swapping (null modem) required, so a 9-pin DB9 M/F straight-through cable is required. The DTE models (-034, -036, -044, -046) require a 9-pin DB9 null modem cable to be used, or more commonly, a straight-through cable with a small null modem adapter.

**Step 3:** Change the serial settings of the *PS2PROX Config* program to match the current configuration of the PS2PROX (default from the factory is 9600,N,8,2).

**Step 4:** The *PS2PROX Config* program is very easy to use. First choose a string-length from the drop-down box. The longer the string, the fewer key overrides there are available so choose the shortest string length needed.

**Step 5:** Click the cursor in a Keyboard Key box and hit the key you wish to override. Then place the cursor in the Output String box to its right and enter the desired output string for this key. You can type ASCII characters into this box .. and you can even enter non-ASCII, 8-bit non-printable characters by enclosing the hexadecimal value of the character in { } brackets. So a Line Feed character can be entered as {0A}. Note that the default key output is completely replaced by your string. So if you leave the Output String box empty, there is no output for the specified key. If you want the key to output its default value followed by a Line Feed you would have to enter the default output character then the Line Feed character. For example to make the Enter key output CRLF, you would enter {0D}{0A} as the Output String. Notice that each {xx} sequence represents just one character of the possible number available to you. (Do not enter {00} – this represents an ASCII NULL characters and is a reserved character.) Special Keys:

- **TAB key** -- To override this key you must press CNTRL-TAB and ‘TAB’ will appear in the Keyboard Key field. ( To enter it in an Output String it is better to use {09} since blank space is impossible to visually distinguish from multiple space-characters.)
- **Insert, Home, PageUp, Delete, End, PageDown, Enter, All Arrow keys ...**  
You can override these keys independently from their duplicate keys that reside in the numeric keypad. To override Insert for instance, press the Insert key once and *Insert* will appear in the Keyboard Key box. Press it a second time and *KP Ins* (Numeric Keypad insert key) will appear in the box. So your selections toggle between the normal key and the Keypad version of that key which allows you to make separate override entries for these keys.

**Step 6:** To send this configuration to the PS2PROX simply click the Send button. A checkmark will appear next to each of the override entries (even if empty) as the written value is auto-verified after writing. You can SAVE your configuration as a file for later recall when you need to program more devices.

**Step 7:** At any time you can verify that the key outputs stored in the PS2PROX match your on-screen/file-stored values. For example if you are unsure if you configured a PS2PROX, you can either

- a) Click the “Read Existing Key Outputs’ button which will load the screen with the key outputs stored in the PS2PROX so you can verify them visually – or—
- b) Open a file that you previously SAVED using the FILE menu which loads those values to the screen, then click the ‘Verify’ button which will compare values stored in the PS2PROX with those onscreen. **Green** checkmarks appear next to those that match, and **Red** question marks appear next to those which do not match. Place your cursor over a question mark for details on a mismatch.

### Additional Configuration Notes:

You may assign a key-output to any of the available-positions on the screen , leaving any or all others empty. So for instance you can add an entry to the # 5 position only, or to #2, and #9 only. There is no need to fill them sequentially or consecutively. PS2PROX simply scans through all entries for every keyboard key press and looks for a matching key to override in any position. The first match it finds will be used, and if no match is found then the currently pressed key is not overridden, and will output the default character (the one printed on the key cap).

## Alternate Verification of PS2PROX Custom Character/String Outputs

After configuring the PS2PROX, you can confirm that correct output characters/strings are being generated by using another Versalent Utility called Simple Term which is also available for download from <http://www.versalent.biz>

With the PS2PROX still connected to the PC COM port, close the *PS2PROX Config* program, start *Simple Term* and change its COM settings to match the PS2PROX. With a PS2 keyboard now connected to the PS2PROX, press the keys which have been overridden and you will see the characters/strings previously programmed. Notice that non-printable characters appear as hexadecimal values enclosed by [] brackets, and the special control characters such as NUL, ESC, SOH, BEL etc appear as [NUL], [ESC] etc. 8-bit characters which have no such ASCII defined symbols appear as hexadecimal values enclosed by [] brackets, i.e. [97] .

### Document Revision Record:

Revision #	Revision Date	Description
V1.00	Mar 16, 2018	Modified from the PS2PRO manual