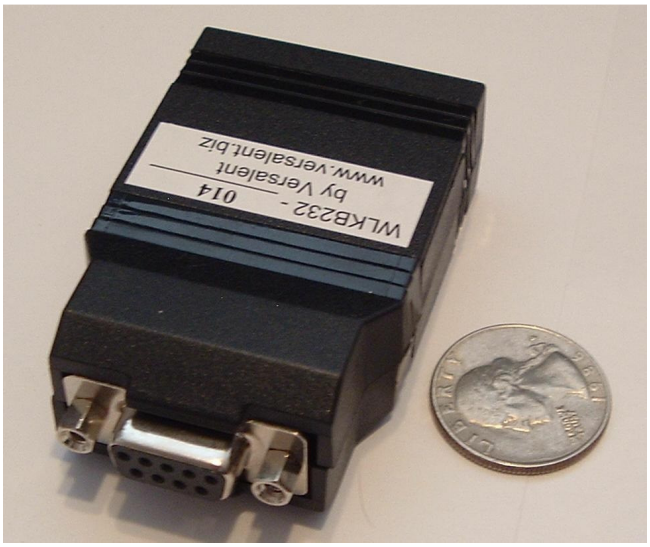
Versalent

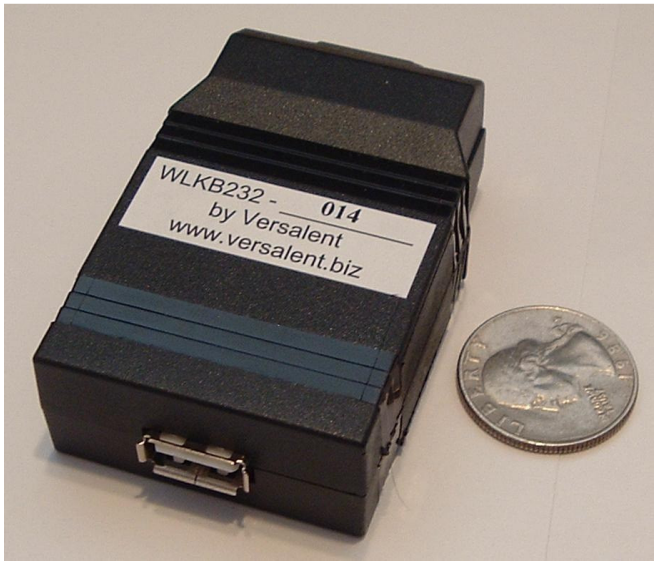
www.versalent.biz

June 2018

WLKB232 & WLKB232V2 USB Wireless Keyboard Serial Converter Manual

Version 2.00
Revised June 27, 2018





General Description

WLKB232 is a small module (2.6" x 1.7" x 0.8") that allows a standard wireless USB keyboard to be used with non-USB RS-232 serial systems. The characters printed on the keys are output as serial characters so that non-PC based hosts can easily use wireless keyboards. USB is simple for users but contains complexities that require specialized software and hardware for even simple keyboard communications. WLKB232 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 offer USB. It brings wireless keyboards easily to the smallest of embedded systems through any standard UART.

The 'V2' version is an upgrade to the previous no-suffix version. It adds the Numlock Override feature which allows users to define an alternate action for the Numlock key which was previously unavailable. If the user turns ON the Numlock Override feature in the Configuration Program (v2.00 and above), then the numeric keypad will revert to 0-9 and arithmetic keys even if some of the keypad keys have custom strings defined. (In the un-suffixed version custom strings were ALWAYS in effect regardless of the state of Numlock, so a full numeric keypad was not available if any of the keypad keys had custom strings defined).

The remainder of this manual describes the operation of both the WLKB232 and all descriptions are also valid for the WLKB232V2 unless otherwise noted.

WLKB232 accepts the USB dongle of many keyboards/mice combos providing the complex USB interactions including power management, enumeration sequence and report exchanges required to 'talk USB'. By default it also translates the USB keycodes to the ASCII character printed on the key so all you have to do is plug it in and serial characters arrive. A companion WLKB232 Configuration Program is provided which allows any key output to be reprogrammed to send up to a 10-character string. These custom strings are stored in the WLKB232's internal non-volatile memory so they are retained during power-down. It transparently manages the CAPS Lock and NUM Lock states so the keyboard sends the modified key cases and operates as expected. And if the keyboard has LEDs it manages those as well. It recognizes and acts on all keys including function keys, arrow keys, keypad keys with Numlock/Ctrl/Alt both on and off etc. It even works with wired USB keyboards converting their output to serial as well.

Wireless keyboard power is provided through its batteries or solar cells. But the dongle must be (5 volt) powered from its USB host connector, and the radio transceiver current requirements of a typical dongle are typically 30mA or more. This is well beyond the ability of 'RS232 power-stealing' techniques to provide, therefore external power for the WLKB232 is required. Various power options allow you to supply the 5V power for both the WLKB232 (and attached USB dongle) via the DB9 connector, or with the optional 'power brick' wall supply. And there are several models which provide DTE or DCE DB9 connection options so you won't need a null-modem adapter to adjust connector signal/pin positions.

When the WLKB232 is powered and operating a small red LED flashes very quickly initially before a USB keyboard/composite device is detected. Then after a recognized

USB device is identified and successfully initialized, the flash rate slows to every ½ second (a noticeably slower rate). So the LED provides visual indicators of applied power as well as the state of the USB connection. Note that WLKB232 works with wired or wireless ***keyboards only*** – it is not a general purpose USB host like a desktop PC and will not operate with any other type of device or USB hub. WLKB232 has very limited USB resources compared to a full USB desktop host and cannot handle the large complexity and flexibility of a full host which recognizes different types/classes of devices.

Reliability Features:

WLKB232 implements an internal watchdog timer, and an internal brown-out detector. If the internal 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. The USB dongle does not have to be unplugged /replugged... the WLKB232 will power-up and re-initialized the USB connection and begin operating. If the supplied power droops 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.

The WLKB232 disables the USB power to the receiver dongle with any disruption as above so the dongle will get reset, however the keyboard will not be reset since it operates on mobile power. The keyboard may require a manual reset if it does not resume normal operation after the dongle drops out, then resumes communication.

*Note that the WLKB232 connector is female even for the optional DTE pin configurations which are generally a male – like those on a (DTE) PC COM port. If male is needed, a small 'gender-bender' adapter is recommended.

USB Wireless Radio Signal USB Dongle WLKB232 **Your product w/RS-232**

Keyboard

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 with a little inward pressure while twisting the blade. After the first internal latch releases, the others release even easier and the two halves of the case separate.

WLKKB232 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. 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 WLKB232 must be powered off/on to reset and accept the new serial settings.

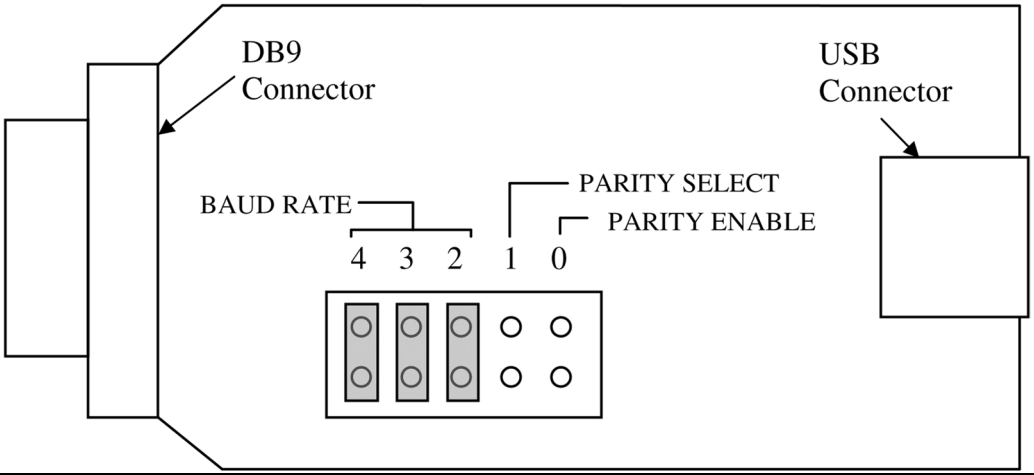
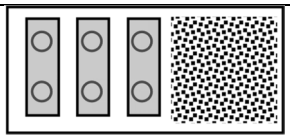
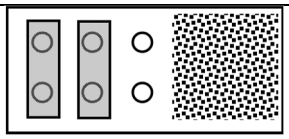
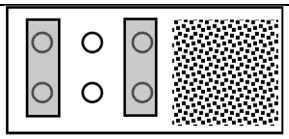
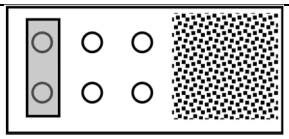
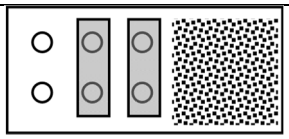
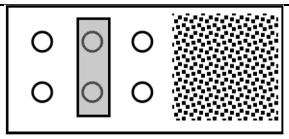
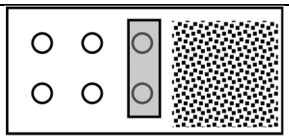
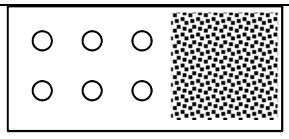


DIAGRAM	BAUD RATE	PARITY
	600 bps	See below
	1200 bps	See below
	4800 bps	See below
	9600 bps	See below
	19.2k bps	See below
	38.4k bps	See below
	57.6k bps	See below
	115k bps	See below

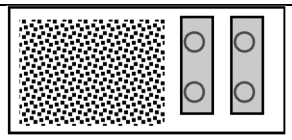
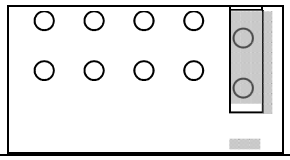
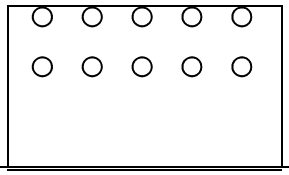
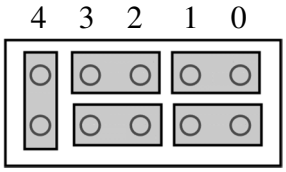
	See above	Parity Enabled, ODD Parity
	See above	Parity Enabled, EVEN Parity
	See above	Parity Disabled

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. WLKB232 devices are shipped in the following default configuration .. 9600 baud, no parity. (All the horizontal blocks have no effect; the only effective one is the vertical one to the far left.)



RS-232 Signal Compatibility:

The WLKB232 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 device 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 WLKB232 requires +5VDC to operate and it provides this power to the USB port for the USB dongle/transceiver. There are two ways to apply power to the WLKB232:

1) Apply power to a pin of the DB9 connector
2) Plug a 'power brick' into the WLKB232

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 @ 50mA available you can select one of the 5VDC input models. If unavailable, or it is more convenient to power the WLKB232 with a wall supply, another option is a 6-12VDC model. The internal regulator in the WLKB232 will then supply the needed +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 unused RS232 pins cannot supply sufficient power for a USB radio transceiver and the WLKB232 – some type of external power is required.

WLKB232-xxx MODELS AVAILABLE

Model#	Features
WLKB232-014	DB9 = DCE, +5VDC power applied to DB9 Pin 4
WLKB232-016	DB9 = DCE, +5VDC power applied to DB9 Pin 6
WLKB232-024	DB9 = DCE, 6-12VDC power applied to DB9 Pin 4 or use optional 6V center-positive wall supply
WLKB232-026	DB9 = DCE, 6-12VDC power applied to DB9 Pin 6 or use optional 6V center-positive wall supply
WLKB232-034	DB9 = DTE, +5VDC power applied to DB9 Pin 4
WLKB232-036	DB9 = DTE, +5VDC power applied to DB9 Pin 6
WLKB232-044	DB9 = DTE, 6-12VDC power applied to DB9 Pin 4 or use optional 6V center-positive wall supply
WLKB232-046	DB9 = DTE, 6-12VDC power applied to DB9 Pin 6 or use optional 6V center-positive wall supply

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 PCs Tx signal connects to the WLKB232 Rx signal, and the PCs Rx signal connects to the WLKB232 Tx signal with a standard straight-thru M/F DB9 cable.

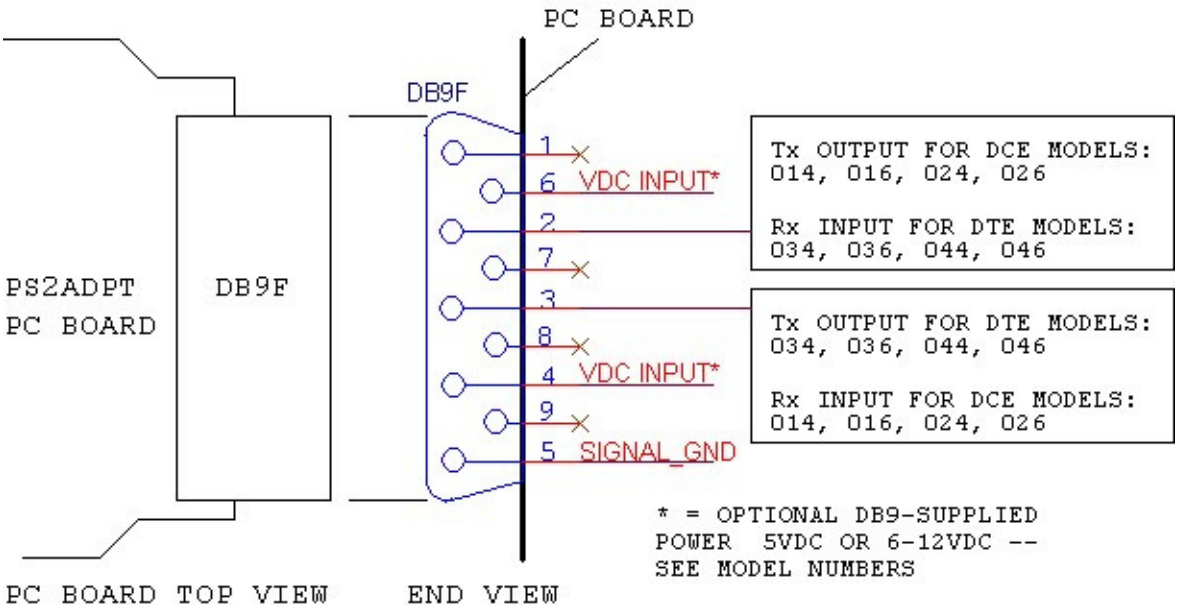
Power Considerations:

USB ports are sometimes used for battery chargers, or lighting power sources and the WLKB232 USB port should not be used for any of these types of devices. It should not be used with any USB device that requires more than 200mA. Typical keyboard dongles require only a fraction of this available power, but as mentioned previously, RS232 ports do not offer sufficient power the WLKB232 and the USB RF receiver. Power must be applied to either a pin of the DB9 connector, or from an external wall-adapter.

WLKB232 Physical, Electrical, Environmental Specifications

Physical:

Size:	2.6" X 1.7" X 0.8"
Weight:	2.6 oz
USB Connector:	Standard Type-A
Host Connector:	Standard DB9 (female)
Case Color:	Black (Ivory available special order)
Power Connector:	5mm X 2.1 mm (Ctr Positive)



WLKB232 DB9 (Female) Pin Assignments Diagram

Electrical:

Absolute Maximum Input Voltage	+15VDC (Internal Regulator Models only)*
Nominal Voltage Input:	+5VDC +/- 5% or 6-12VDC (per model number)
Current Input:	10mA maximum plus USB dongle 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 WLKB232 is designed for simplicity – it generally outputs a single ASCII character (the one printed on the keyboard key) when the key is pressed. Or if you have reprogrammed the key output it sends the string of your choice. If the key is held then the WLKB232 auto-repeats the serial character or string output. Although the keyboard/dongle generate USB keycodes, the WLKB232 does not. It only sends the serial-character information you are interested in.

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 WLKB232 sends 8-bit characters to identify these 'non-printable' keys. And to support the use of the ALT key as a character-modifier the WLKB232 sends two characters to identify ALT-x keys. Please refer to http://www.versalent.biz/manuals/wlkbmap.pdf which is a diagram of a 102-key PC keyboard showing all key positions, and the associated character(s) which the WLKB232 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 CRTL pressed
5) Key pressed with NUMLOCK active

You will notice that the NUMLOCK, 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. CAPSLOCK does send a code and then continues to act as a modifier when active.

Environmental:

Max Operating Temperature:	+85ºC
Min Operating Temperature:	0ºC
Max Storage Temperature:	+100ºC
Min Storage Temperature:	-40ºC
Humidity:	Non-condensing at all temperatures

How to Configure WLKB232 for Custom Character/String Output

Note: Any key reprogrammed to output a single ASCII character is subject to the CTRL, ALT, CAPS LOCK, SHIFT KEY modifications. The output of a key reprogrammed to send a string (more than one character) is not altered by any of these 'modifier' keys.

NOTE: During the following programming process, all keypresses described are to be executed on the keyboard which is connected to your PC. The Configuration Program is running on your PC and all its keyboard input comes from the PC's keyboard. Ignore the WLKB232's wireless keyboard until the programming sequence is complete.

1) Download and install '*WLKB232 ConfigV2* Windows program. It programs both the older WLKB232 as well as the WLKB232V2 with the Numlock Override feature.
2) Connect WLKB232 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 WLKB232 DB9 connector while connected to the COM port for programming.]**
3) Set COM port to same serial communication settings as the WLKB232
4) For any number of keys, click the mouse in the Keyboard Key box and press the single keyboard key for which you want a custom output. Immediately to its right enter the character string to be output instead of the default character output.
5) Click 'the Add To List' button. Repeat steps 4 & 5 to alter any number of keys. You cannot add duplicates .. that is, once you enter a string for the F1 key .. you cannot enter another without first deleting the previous one from the list.
6) For the WLKBV2 device you can select the Numlock Override mode from the top menu. When ON, devices will be programmed to revert the keypad to numerics when the Numlock key is active even if keypad keys are reprogrammed with custom strings. When OFF, devices will be programmed to operate like the non-suffixed version – custom strings defined for keypad keys are always in effect no matter what the state of the Numlock key.\
7) After entering all the output strings desired, you can save this list to a file for later recall, or just click SEND STRINGS TO WLKB232.
8) The list will be sent and programmed into non-volatile flash memory which is retained until you again use this program to change it. Note that even if you reprogram just one key, the configuration program must send the entire string set to the WLKB232 because the erase phase of the update process erases the entire block of stored strings. Any keys you do not override are by default filled with their default characters which are transmitted to the WLKB232. So it may appear that there is a lot of communication for a small number of key strings, but most of it will typically be the transmission of the default characters.

.

More Details For Each Step Above:

Step 1: The *WLKB232 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 *WLKB232 Config* program to match the current configuration of the WLKB232 (default from the factory is 9600,N,8,2).

Step 4: The *WLKB232 Config* program is very easy to use. You can override the default output for keyboard keys. Place 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}. 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 (out of the maximum of 10) in the output string. (Do not enter {00} – this represents an ASCII NULL character and is a reserved character.)

Step 5: To send this configuration to the WLKB232 simply click the Send Key Strings button. The entire array of strings will be updated in about 15 seconds. When it completes an auto-verify cycle runs to readback all the values to verify that the programming completed successfully. Below the Communications Details window a green checkbox will appear and a message indicating that all the strings were verified successfully, or a red question mark and text indicating an error.

Step 6: At any time (without clicking Send Key Strings) you can verify that the key outputs stored in the WLKB232 match your on-screen values (perhaps retrieved from a file or entered manually). For example if you are unsure if you already configured a WLKB232, you can either

a) Click the "Read Existing Key Strings' button which will load the screen with the key outputs stored in the WLKB232 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 Key Strings' button which will compare those stored in the WLKB232 with those onscreen. A Green checkmark appears if all compare successfully, and Red question mark appears if any fail. And as mentioned previously, any keys you do not specify in your list assume their default character values, and even those default characters are verified during the process.

Additional Configuration Notes:

There is no need to enter keys in any particular order. The configuration program simply scans through your entries and re-programs the keys you have entered (and programs any not in your on-screen list with the default characters). Note that it re-programs every key even if you update only one. So you have to enter all your overrides in a single list – you cannot alter 20 keys, then incrementally alter just one key (without re-specifying the other 20). So it is a good idea to save your key string sets to a file since it is much easier to recall a list, then add or alter one string than to re-enter your whole list. Any key not in your override list will be re-programmed to its default value when you hit SEND STRINGS TO WLKB232 .

To aid your reprogramming efforts, you can SAVE multiple sets of key strings to disk. Then if you decide you need to add one more to a list.. you can recall the original list from a file, add your new entry, reprogram the WLKB232, and again save to disk for future reference. Or you can first retrieve all the current keystrings from the WLKB232 using the Read Key Strings button – all the non-default ones will appear in the on-screen list, then you can alter the list as needed and send it back to the WLKB232.

Reprogramming the numeric keypad section of the keyboard has some restrictions. Since the PC does not distinguish between the numeric-keypad 7-Home key, and the Home key near the PageUp/Dn keys, you cannot reprogram the non-numlock state of the numeric keypad keys without some interaction with the Home key. Typically to reprogram the numeric keypad keys you will have to turn on NumLocks during the programming process.

Verifying WLKB232 Custom Character/String Outputs

After configuring the WLKB232, you can confirm that during actual operation, the correct output characters/strings are being generated for the specified keys by using another Versalent Utility called Simple Term which is also available for download from http://www.versalent.biz

With the WLKB232 still connected to the PC COM port, close the *WLKB232 Config* program, start *Simple Term* and change its COM settings to match the WLKB232. With a keyboard or dongle connected to the WLKB232, type the keys which have been overridden and you will see the characters/strings which were just programmed. Notice that non-printable characters appear as hexadecimal values enclosed by [] brackets, and the special control characters such as ESC, SOH, BEL etc appear as [ESC] , [SOH] etc. 8-bit characters which have no such ASCII defined symbols appear as hexadecimal values enclosed by [] brackets, i.e. [97] representing 0x97.

Document Revision Record:

Revision #	Revision Date	Description
V1.00	Jan 3, 2011	Initial Release
V1.01	Feb 14, 2011	Add programmability description
V1.02	May 6, 2011	Fix minor typographical errors
V1.03	March 30, 2012	Add outline around baud jumper block to orient user
V1.04	May 13, 2012	Change connector label in diagram from 'PS2' to 'USB
V1.05	June 1, 2012	Add note that RS232 cannot power the device.
V2.00	June 26, 2018	Add description of Numlock Override feature and the 'V2' suffixed version.