

# Versalent

[www.versalent.biz](http://www.versalent.biz)

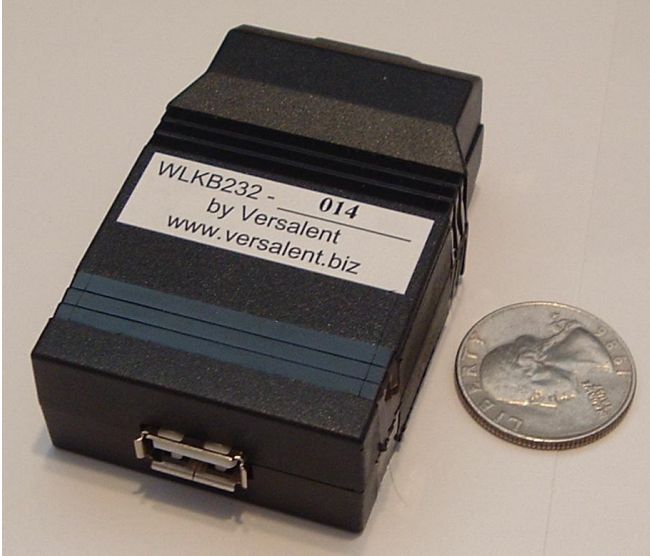
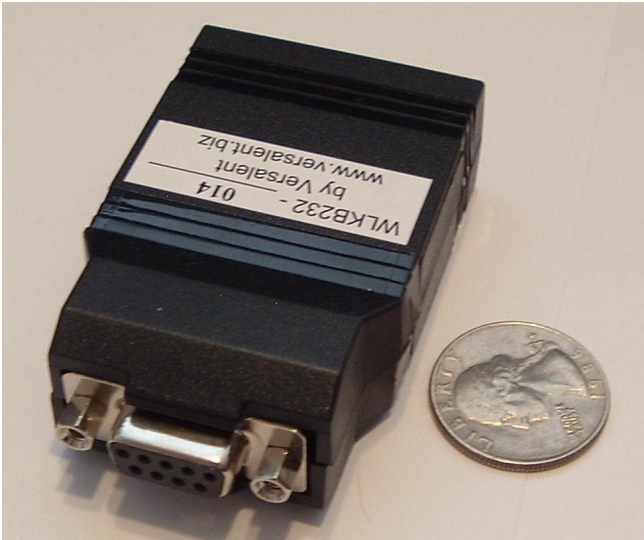
Sept 2025

---

## WLKB232 & WLKB232V2 USB Wired/Wireless Keyboard Serial Converter Manual

Version 3.03

Revised Sept 23, 2025



## Table of Contents

General Description.....	3
V2 Version (with NUMLOCK feature) .....	3
LED Indicator.....	4
Keypress Output.....	4
Reliability Features .....	5
Baud Rate/Parity Control .....	5
E-Baud Feature.....	8
Extended Function Keys (F13-F24).....	9
RS-232 Signal Compatibility .....	9
Power Inputs.....	10
Applying Power Through the DB9 Connector.....	10
USB Port Power .....	10
WLKB232-xxx Models Available .....	11
WLKB232 Specifications .....	12
Physical .....	12
WLKB232 DB9 (Female) Pin Assignments Diagram .....	12
Electrical Specifications.....	12
Environmental Specifications .....	13
Configuring Output Strings.....	13
Verifying WLKB232 Character/String Outputs .....	15
Document Revision Record .....	15

## General Description

WLKB232 is a small module (2.9" x 1.7" x 0.8") that allows a standard wired/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 providing a standard RS-232 output which can interface to almost any computer system – even small micros that do not offer USB. It brings wireless keyboards easily to the smallest of embedded systems through any standard UART.

### V2 Version (with NUMLOCK feature)

The 'V2' version is an upgrade to the original version. It makes the keypad obey the NumLock key **when keypad keys have been user-programmed.** (Original version ALWAYS outputs user-programmed values regardless of the state of NumLock – so a full numeric keypad is not available if keypad keys have been re-programmed). V2 devices are backward compatible with the original version if the NUMLOCK feature is off.

The remainder of this manual describes the operation of the original WLKB232 and the V2 version 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 be a string of 0 to 10-characters. 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 also works with wired USB keyboards.

Wireless keyboard power is provided through its batteries or solar cells but the dongle must be (5 volt) powered from its USB host connector. Radio transceiver current requirements of a typical dongle are 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 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 no null-modem adapters are needed to adjust connector signal/pin positions.

## LED Indicator

When the WLKB232 is powered and operating a small red LED flashes very quickly initially before a USB keyboard or dongle 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.

## Keypress Output

The WLKB232 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 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. The keyboard/dongle generates USB keycodes and the WLKB232 translates those to serial characters/strings.

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 CTRL 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.

## Reliability Features

WLKB232 implements an internal watchdog timer, and an internal brown-out detector. If the internal microcontroller gets 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.

\*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 Keyboard	Radio Signal	USB Dongle Receiver	WLKB232	Your products RS232 Serial Port
--------------------------	-----------------	------------------------	---------	------------------------------------

### Typical Application

## Baud Rate/Parity Control

Baud and parity are set using the E-Baud setting (case does NOT need to be opened), or 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 horizontal 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.

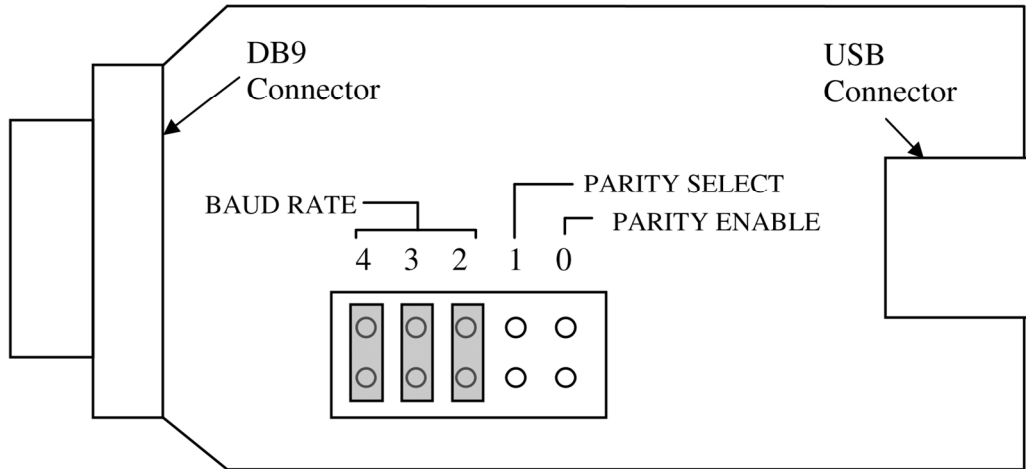
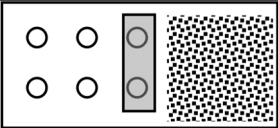
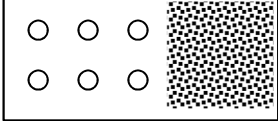
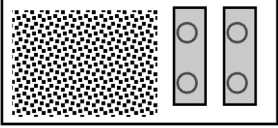
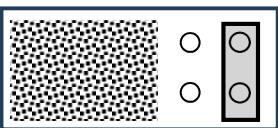
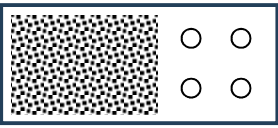
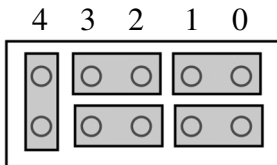


DIAGRAM	BAUD RATE	PARITY
	600 bps	See below
	1200 bps	See below
	4800 bps	See below
	9600 bps	See below
	19.2k bps	See below
	38.4k bps	See below

	57.6k bps	See below
	115.2k bps	See below
	See above	Parity Enabled, ODD Parity
	See above	Parity Enabled, EVEN Parity
	See above	Parity NONE

**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.)



## E-Baud Feature

Devices with model numbers ending in ‘E’ (WLKB232-014E, WLKB232-024V2E etc) include the E-Baud feature which allows the baud rate and parity to be set electronically – without opening the case. The baud rate can be set from 600 to 115.2k by issuing the following RS232 command:

### **/BWLKB232xy**

- /B is the E-Baud command
- WLKB232 is a fixed ‘key’ which prevents inadvertent baud rate changes
- x is one ASCII character that defines the baud rate ( see table below)
- y is one ASCII character that defines the parity (see table below)

This command can be issued by the WLKB232 Configuration application, or even manually with the use of a terminal emulation program.

	<b>Baud Rate</b>		<b>Parity</b>
x = ‘0’	Use internal jumper settings	y=‘0,1 or 2’	Per jumpers
x = ‘1’	600	y=‘0’ y=‘1’ y=‘2’	ODD NONE EVEN
x = ‘2’	1200		
x = ‘3’	4800		
x = ‘4’	9600		
x = ‘5’	19.2k		
x = ‘6’	38.4k		
x = ‘7’	57.6k		
x = ‘8’	115.2k		
x = ‘9’	2400		

The E-Baud command is issued at the device’s initial baud rate, and it responds with “SUCCESS” also at that initial baud rate. The baud rate/parity then changes and operation resumes with no reset or power cycling required. Because the new values are stored in non volatile memory, the WLKB232 will then power up at this baud rate.

Notice that when x=0, the device reverts to using the internal jumpers for its baud/parity settings. This is the ‘compatibility’ mode which is the factory default, and makes the device operate like the original non-E versions. When x > 0 these jumpers may be in any positions since they are ignored.

With E-Baud devices there is no way to visually confirm the unit’s current baud/parity settings. The WLKB232 Configuration app includes a ‘Find WLKB232’ menu item that searches all serial ports, at all baud rates and all parity settings – sending a request for the firmware version (/V command) and awaiting a valid ‘WLKB232vX.XXE’ response.

## **Extended Function Keys (F13-F24)**

Initial versions of the WLKB232 supported the standard 102-key keyboards, providing 96 programmable keys, and 8 ‘modifier’ keys like CTRL and ALT etc. which generate no output. An ‘F’ option has been added which expands the support function keys to include F13-F24 so the number of programmable keys is increased to 108. These keys are typically available on some industrial type keyboards. Models with the ‘F’ suffix include this upgrade as well as E-baud, and the V2 NumLock feature which returns the keypad to numerics even if keypad keys have been reprogrammed.

## **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 Inputs

The WLKB232 requires +5VDC internally to operate and to power the USB port. 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 (power jack available in some models)

### Applying Power Through the DB9 Connector

Power can be applied to either Pin 4, Pin 6, or Pin 9 of the DB9 depending on model. See table below. If your system has +5VDC @ 50mA available you can select one of the 5VDC models. If +5V is unavailable, or it is more convenient to power the WLKB232 with a wall supply, other options include the 'legacy' 6-12VDC models with DC power jack, or the newer 3V-30V models also with a power jack . (Internal regulator provides the needed +5VDC.) The 6-12V, or 4-30V can also be supplied via the DB9 connector.

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 pin-swapping adapters.

### USB Port Power

USB ports are sometimes used for battery chargers, or lighting power sources. The WLKB232 USB port should **not** be used for any of these functions. It should not be used with any USB device that requires more than 200mA (100mA for 4-30V models). Typical keyboards or wireless dongles require a small fraction of this available power.

\*\*Note that unused RS232 pins/signals *cannot* supply sufficient power (using 'RS232 power-stealing' techniques) for the WLKB232 USB host – external power is required.

## WLKB232-xxx Models Available

Legacy Model#	Features
WLKB232-014, -014E	DB9 = DCE, +5VDC power applied to DB9 Pin 4
WLKB232-016, -016E	DB9 = DCE, +5VDC power applied to DB9 Pin 6
WLKB232-024, -024E	DB9 = DCE, 6-12VDC power applied to DB9 Pin 4 or via 6-12V center-positive wall supply jack
WLKB232-026, -026E	DB9 = DCE, 6-12VDC power applied to DB9 Pin 6 or via 6-12V center-positive wall supply jack
WLKB232-034, -034E	DB9 = DTE, +5VDC power applied to DB9 Pin 4
WLKB232-036, -036E	DB9 = DTE, +5VDC power applied to DB9 Pin 6
WLKB232-044, -044E	DB9 = DTE, 6-12VDC power applied to DB9 Pin 4 or via 6-12V center-positive wall supply jack
WLKB232-046, -046E	DB9 = DTE, 6-12VDC power applied to DB9 Pin 6 or via 6-12V center-positive wall supply jack
Newer Model #	
WLKB232-114, -114E	DB9 = DCE, 4-30VDC power applied to DB9 Pin 4, or center-positive wall supply jack
WLKB232-116, -116E	DB9 = DCE, 4-30VDC power applied to DB9 Pin 6, or center-positive wall supply jack
WLKB232-134, -134E	DB9 = DTE, 4-30VDC power applied to DB9 Pin 4, or center-positive wall supply jack
WLKB232-136, -136E	DB9 = DTE, 4-30VDC power applied to DB9 Pin 6, or center-positive wall supply jack

(All available with V2 suffix providing more flexible operation of the numeric keypad)

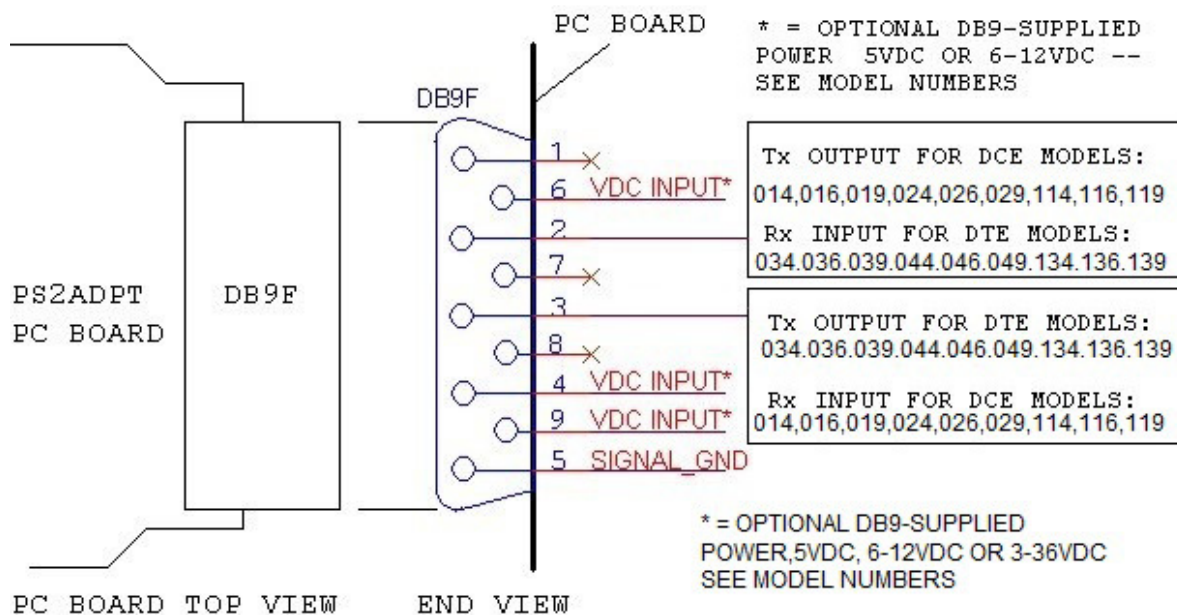
**TABLE 2**

*Note that the COMx ports on a standard PC are configured as DTE its RS-232 signals mate directly to the DCE models in the table above with no pin swapping.*

# WLKB232 Specifications

## Physical

Size:	2.9" X 1.7" X 0.8"
Weight:	1.4 oz
USB Connector:	Standard Type-A
Host Connector:	Standard DB9 (female)
Case Color:	Black (Ivory available special order)
Power Connector:	5.5 mm X 2.1 mm (Ctr Positive)



## WLKB232 DB9 (Female) Pin Assignments Diagram

## Electrical Specifications

Absolute Maximum Input Voltage	5V MODELS: +5.25VDC 6-12V MODELS: +15VDC 4-30V MODELS: +32VDC
Nominal Voltage Input:	+5VDC, 6-12VDC, 4-30VDC +/- 5%
Max USB Device Current:	5VDC power : 200mA 6-12VDC power : 100mA 4-30VDC power : 100mA
Current Input (5V, 6-12V models)	35mA max plus USB keyboard/dongle current

Current Input (4-30V models)	400mA surge, 120mA continuous
Baud Rate:	600 to 115.2k selectable via jumpers or E-Baud
Parity:	Selectable ODD/EVEN/NONE
DB9 Connector Configuration:	DTE or DCE (per model number)

## **Environmental Specifications**

Max Operating Temperature:	+60°C
Min Operating Temperature:	0°C
Max Storage Temperature:	+70°C
Min Storage Temperature:	-40°C
Humidity:	Non-condensing at all temperatures

## **Configuring Output Strings**

### **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 refer to your computer’s keyboard. The WLKB232’s keyboard/dongle should remain unplugged until the programming sequence is complete.**

- 1) Download and install ‘*WLKB232 Config*’ Windows program. It programs all versions of the WLKB232.
- 2) Connect the 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 either a PROGIT-1 Programming Adapter, or a power-tap cable adapter to allow the application of DC power to the WLKB232 DB9 connector.]** Also DTE models (-034, -036, -044, -046,-134, -136) require a null-modem (pin swapping) adapter if a PROGIT-1 is not used.
- 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 – a keyboard appears on screen. Select the key for which you want a custom output. Immediately to its right enter the character or string to be output.
- 5) Click ‘the Add To List’ button. Repeat steps 4 & 5 to alter any number of key outputs. You cannot add duplicates – only one entry per key.

- 6) For the WLKBV2 device you can select the Numlock Override mode from the top menu. When ON, devices will revert the keypad to numerics when the Numlock key is active even if keypad keys are programmed. When OFF, devices will operate like the non-suffixed version – custom strings defined for keypad keys are *always* in effect regardless of 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, and/or just click SEND STRINGS TO WLKB232.
- 8) The list will be programmed into non-volatile flash memory which is retained until changed. Note that even if you reprogram just one key, the configuration program updates the entire key-set. (Keys not in your list are programmed with their default characters.)

### **More Details For Each Step Above:**

**Step 1:** The *WLKB232 Config* program is available from the DOWNLOADS section of the Versalent website: <https://www.versalent.biz?P=dl.htm>

**Step 2:** The DCE models (-014, -016, -024, -026, 114, -116) 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/adaptor to be used.

**Step 3:** Serial settings for the *WLKB232 Config* app and the WLKB232 device must match. They can be set manually or by using the ‘Find WLKB232’ menu item.

**Step 4:** The *WLKB232 Config* app is very easy to use. You can override the default output for any number of keyboard keys. The Output String accepts ASCII characters.. you can enter non-ASCII, 8-bit characters by enclosing their hexadecimal value in {} brackets. So a Line Feed character can be entered as {0A}. Entering a key but leaving the Output String box empty causes no output for that key. Examples: 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 maximum of 10) in the output string. (Do not use {00} – this is an ASCII NULL character and is a reserved.)

**Step 5:** The Send Strings button updates the WLKB232 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 you can verify that the key outputs stored in the WLKB232 match your on-screen list (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 current WLKB232 key outputs for visual verification – or—

- b) Open a file previously SAVED using the FILE menu (loads those values to the screen), then click the ‘Verify 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 not specified in your list assume their default character values, and even those default characters are verified during the process.

## Additional Configuration Notes:

Programmed keys can be entered in any order. Since all keys are programmed as a single block, no incremental updates are allowed -- all your key entries must be in a single list. So it is a good idea to save any key string sets to a file for easy recall/edit.

## Verifying WLKB232 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 any terminal emulator such as the Versalent Simple Term which is available for download from <https://www.versalent.biz?P=dl.htm>

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 now connected to the WLKB232, type the programmed keys on the WLKB232’s keyboard and verify the outputs. Non-printable characters 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	Mar 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 Numlock Override feature and the ‘V2’ version.
V2.01	May 14, 2020	Fix a spelling error
V2.02	Aug 8, 2021	Minor changes to structure and grammar.
V2.03	Feb 20, 2022	Correct overall length from 2.6” to 2.9”

V2.04	Jul 12/2023	Add E-Baud feature description
V2.05	Sep 10, 2023	Add max USB current spec
V2.06	Dec 16, 2023	Correct supply connector dimension chg 5mm to 5.5mm
V3.00	Sept 28, 2024	Add 4-30V models
V3.01	Nov 21, 2024	Add TOC, Chg to 4-30V to allow for more keyboard devices
V3.02	Feb 4, 2025	Simplify V2 description
V3.03	Sept 23, 2025	Add F13-F24 keys, and 2400 baud to E-baud settings