# vkb\_led\_jg\_plugin\_db

## Introduction

The VKB Gladiator NXT flight stick has three LEDs: a blue/red in the base, a red near the hat switch, and a RGB (red, green, blue) LED at the top of the handle. These LEDs have programable levels of brightness and speeds of blinking. This Joystick Gremlin plugin allows these LEDs to be activated with buttons on the stick or any other device configured in Joystick Gremlin.

**Warning**: this code interfaces with your VKB device. I have found this code safe to use, but you assume all risks when applying your device.

Although there are only three LEDs, multiple buttons can activate the same LEDs in varying configurations. This plugin uses a database engine to create an external record to track the order of button, and their associated LED, activations and deactivations. This process is described in more detail below.

A database file, *VKB\_LED\_event\_stack.db*, is created in the same directory that holds the plugin. Do not delete or move this file while the Joystick Gremlin configuration is active, but it can be deleted afterward. A new file is created at each JG activation.

## Credit Where Credit Is Due

This plugin could not function without the following fantastic packages and code examples:

* bitstruct written by Erik Moqvist & Ilya Petukhov located at <https://pypi.org/project/bitstruct>
* pyvkb written by ventorvar located at <https://github.com/ventorvar/pyvkb>. The code for the LEDClass, set\_LEDs() and LED\_conf\_checksum() are slightly modified but essentially from this package. My modifications include simplifying the RGB colors to use the VKB 0-7 range and added directives to open and close the VKB device.
* pywinsub written by Rene F. Aguirre located at <https://pypi.org/project/pywinusb>
* SQLite, a public domain database engine, integrated into Python, <https://sqlite.org/index.html>
* vkb-msfs-led by tiberriusteng located at <https://www.github.com/tiberiusteng/vkb-msfs-led>. Mostly used as an example and for inspiration.
* And, of course, Joystick Gremlin by whitemagic located at <https://whitemagic.github.io/JoystickGremlin/>

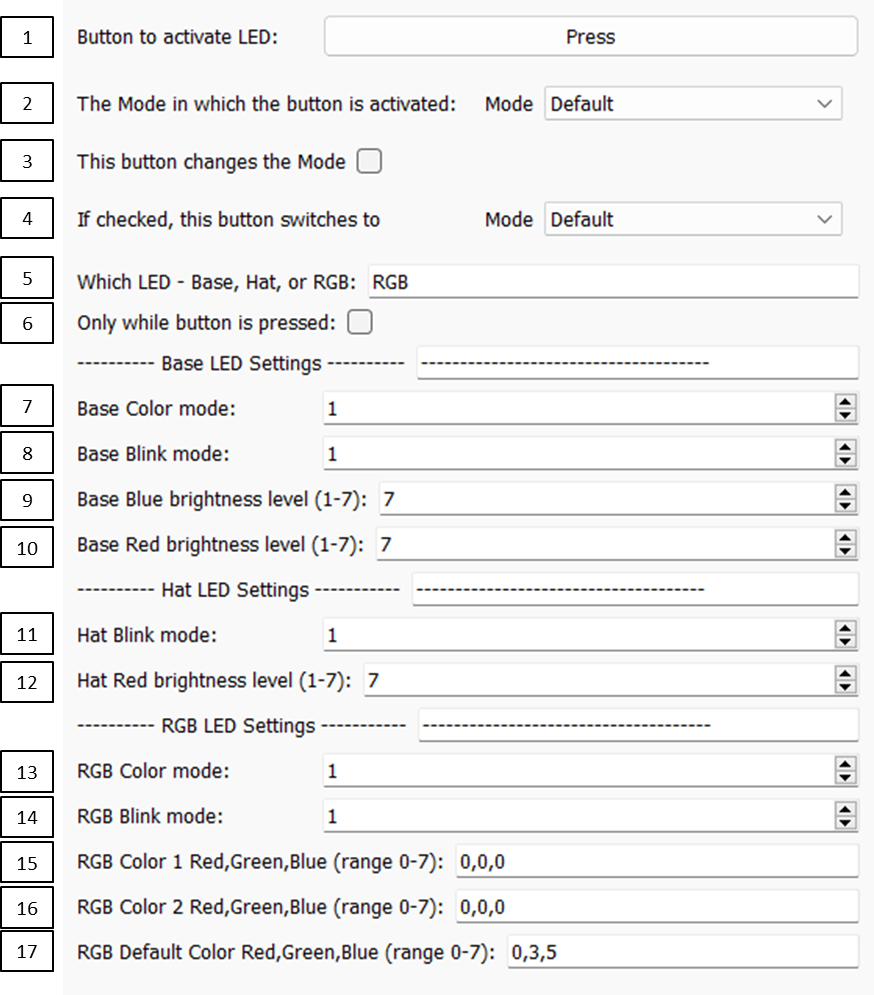
## Installation Instructions

1. Save the bitstruct & pywinsub folders to the Joystick Gremlin programs directory. For me this is: C:\Program Files (x86)\H2ik\Joystick Gremlin
2. Save the two SQLite files (\_sqlite3.pyd and sqlite3.dll) and the one directory (sqlite3) to the same Joystick Gremlin directory
3. Save vkb\_led\_jg\_plugin\_lib.py to the same Joystick Gremlin directory
4. Save vkb\_led\_jg\_plugin.py in a conveniently accessible directory
5. In vkb\_led\_jg\_plugin.py, the product id is set for a right-handed Gladiator NXT. This plugin SHOULD work with a left-handed Gladiator and both left and right-handed Omni throttles as well as older Gladiators that have a base LED (blue and red), a hat LED (red) and a RGB LED. Edit the file with a text editor and enter in the appropriate product id.
6. Open Joystick Gremlin, go to the Plugins tab, and choose Add Plugin, and navigate to where vkb\_led\_jg\_plugin.py is located.
7. Select desired settings as described below.

## Setting Plugin Options

Each instance of this plugin allows one button to activate one LED. However, creating multiple instances of the plugin allow for multiple buttons to activate the same LED (for different functions) or one button to activate multiple LEDs. The actual simulator/game functions are set in Joystick Gremlin and the cosmetics of lights are set here. For example, you can set the RGB LED to flash green when the landing gear are dropped, or set it to blue when switching modes. Each element of the interface is described below.

### vkb\_led\_jg\_plugin\_db interface



1. Assign the LED to a button by clicking “Press” and then pressing the desired button on your device. This can be the VKB flight stick or any other device available in Joystick Gremlin (JG), a throttle for example.
2. Designate the mode that button will be active in. There will be choices only if the JG configuration has multiple modes. For example, the same button could have different functions depending on the ode. Likewise, this same button could have different LED configurations depending on the mode.
3. Indicate whether this button switches modes. In JG a button can be assigned the Switch Mode action. Switching Modes has implications for the order of LED activation and a check here allows the plugin to act accordingly.
4. If the button Switches Modes, indicate the mode being switched to.
5. Select the desired LED by typing in Base, Hat, or RGB. This entry text is not case sensitive but cannot correct misspellings.
6. Check this box if the LED is lit only while the button is pressed, otherwise, the LED will stay “on” until the button is pressed again. This option will not be applied to buttons that switch modes. A button that switches modes can only have an LED on or off.

The three LEDs have different possible configurations. The Base LED can be red, blue, alternating red and blue, or red and blue, with different brightness of color and speed of blinking (or constant). The Hat LED is red only but can vary in brightness and rate of blinking. The RGB LED can be a constant color or alternate between two colors at a given rate of blinking. The colors are defined by setting the red, green, and blue (RGB) components of color. The VKB RGB LED utilizes a simplified color scheme in that each red, green, and blue values can only range between 0 and 7 (instead of the hundreds or thousands of values a computer monitor may have). Because of the vast differences in these three LED configurations, there is a separate section for each LED in the interface. Only the section that applies to chosen LED will be used.

### Base LED Settings

1. Enter a number between 1 and 5 to indicate the color or color combination to use:
   1. Blue
   2. Red
   3. Blue then Red
   4. Red then Blue
   5. Blue and Red
2. Choose the blink rate:
   1. Constant
   2. Slow
   3. Fast
   4. Ultra-fast
3. Enter a brightness level for the blue LED, 1 to 7 where 7 is brightest. Only applies if the blue LED is activated.
4. Enter a brightness level for the red LED, 1 to 7 where 7 is brightest. Only applies if the red LED is activated.

### Hat LED Settings

1. Choose the blink rate:
   1. Constant
   2. Slow
   3. Fast
   4. Ultra-fast
2. Enter a brightness level for the red LED, 1 to 7 where 7 is brightest.

### RGB LED Settings

1. Select the color mode:
   1. Color 1
   2. Color 2
   3. Color 1 then Color 2
   4. Color 2 then Color 1
2. Select the blink rate:
   1. Constant
   2. Slow
   3. Fast
   4. Ultra-fast
3. Define Color 1 using values of 0 to 7 for red, green, and blue.

RGB color values combine to form one color. Some examples:

7,0,0 = bright red

1,0,0 = light red

0,7,0 = dark green

0,0,3 = moderate blue

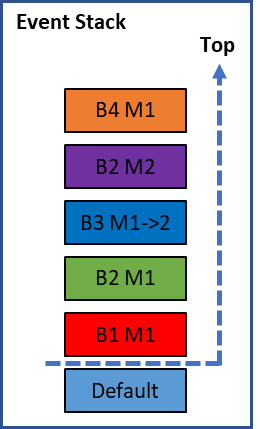
7,0,7 = purple

7,7,0 = yellow

0,7,7 = teal blue

1. Define Color 2 using values of 0 to 7 for red, green, and blue.
2. Define the color to be used when the event stack is empty (no buttons have been pressed). The default, default value is 0,3,5 because that is what the VKB stick use when first plugged in. Use the VKB configuration software to set a different starting value and set this default value to maintain it.

## Event Stack Example

This plugin manages the on/off state of buttons and prioritizes the LED activation through the use of a stack. Note that a true “stack” is typically last-in-first-out but the current process allows items mid-stack to be removed. For example, the figure to the right is a snapshot of the event stack. At the point in time of this snapshot, five buttons have been pressed. B1 changed the LED to red, then B2 changed the LED to green. B3 changed the LED to blue and switched the mode from M1 to M2. B2 has a function in the M2 mode and when pressed changed the LED to purple. Finally, B4 was pressed and change the LED to orange. The top of the stack represents the current color and state of the LED.

The hypotheticals below uses the snapshot to the right as the starting point:

* If B4 was pressed again, it would be removed from the stack and the LED would change to purple.
* If B2 was pressed, the purple entry would be removed but the LED would stay orange.
* If B1 was pressed, the red entry would be removed and the LED stays orange.
* Note that in order to remove the B2 Mode 1 green entry, the Mode 1 would have to be activated. Similarly, to remove B2 M2 purple, Mode 2 would have to be in play.
* If B1 was pressed, the red entry would be removed and the LED stays orange. If B1 was pressed again, a B1 entry would be added to the top of the stack and the LED would change to red.

The purpose of the event stack to have the LED always set to the most recent (top of stack) color, but also remove entries of buttons as they are turned off. If you backed through the button order (B4, B2 M2, B3, B2 M1, B1), the LED would change in that order as well (orange, purple, blue, green, red, default blue).

## Modes

In the examples above, a button that changes a mode switches from Mode 1 to Mode 2 with the first press, then reverses the switch from Mode 2 to Mode 1 with a second press. To accomplish this function you configure the button in JG to Switch Modes to Mode 2 while in Mode 1, and you have to configure this same button to Switch Modes from 2 to 1 while in Mode 2. However, in this plugin you only have to create one instance to allow the toggling of an LED on when switching from Mode 1 to Mode 2, and the LED off when switching from Mode 2 back to Mode 1.

You can also chain mode switching to activate more than 2 modes. For example, in JG you could have a button press in Mode 1 switch to Mode 2; then while in Mode 2 switch to Mode 3. While in Mode 3, the button could switch back to Mode 1 from 3. With the assumption that Mode 1 uses the default color settings, you only have to create two instances of this plugin: the switch from 1 to 2 and the switch from 2 to 3. The turning off of the LED (that is, setting it to the default state) is handled automatically when the button is pressed while in Mode 3.

For example, the default Mode 1 could have basic flying functions, Mode 2 could have air combat functions, and Mode 3 could focus on ground attack functions. The switch from 1 to 2 could activate the LED to slowly blink blue and the switch from 2 to 3 could make the blue blink fast, and finally when switching back to Mode 1 the LED is set to the default color.

There is a JG action to cycle through Modes. If there are more than 2 modes, this plugin will not be able to synchronize LED settings to each mode. This plugin requires the switching action to be attached to each button-mode separately.

## Resetting the Flight Stick

The LED settings stored in the VKB device via VKBDevCFG are overwritten during the plugin's use, but those settings will be restored when the device is unplugged and re-plugged in.

The plugin does not interact with the flight simulator/game in anyway. If LEDs are lit when the simulation or game ends, you will have to push buttons to get back to a neutral state, or simply, unplug and re-plug the device back in.

## Future Developments

If future versions of JG allow dropdown lists, I will use them to make entries easier in the user interface. I also may also replace the hard coding of the product id with a dynamic “button press.”