

HANDHELD
S C I E N T I F I C

BT-500 Bluetooth Adapter for USB keyboard & Mouse

User Manual

Version 5.4.5

Copyright © 2014-2022 by Handheld Scientific, Inc.

All rights reserved.

Email comments/feedback to: support@handheldsci.com

Turning a wired keyboard into wireless, it
doesn't get any simpler than that.

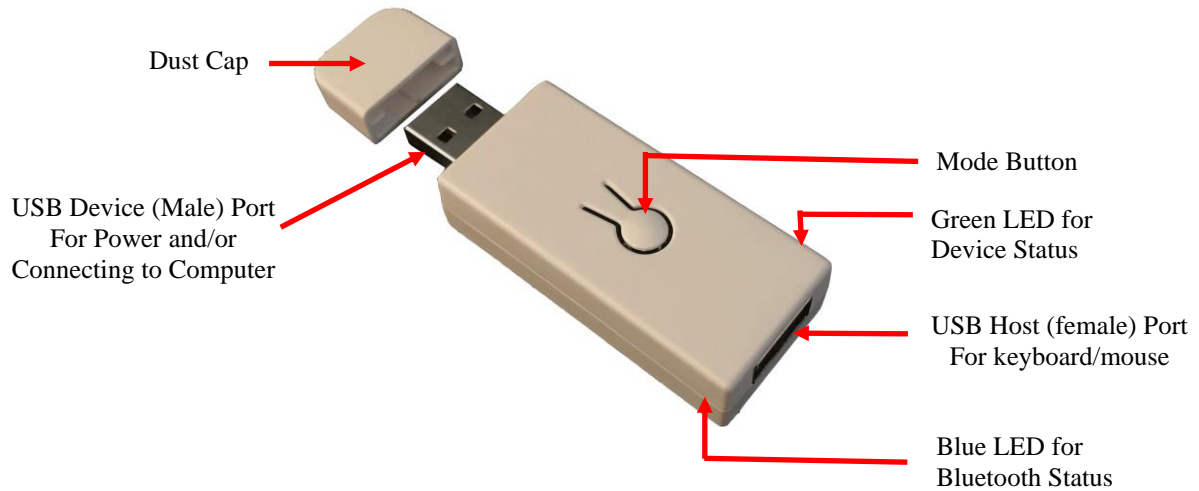
But sometimes, the greatest innovations
are the simplest.

Table of Contents

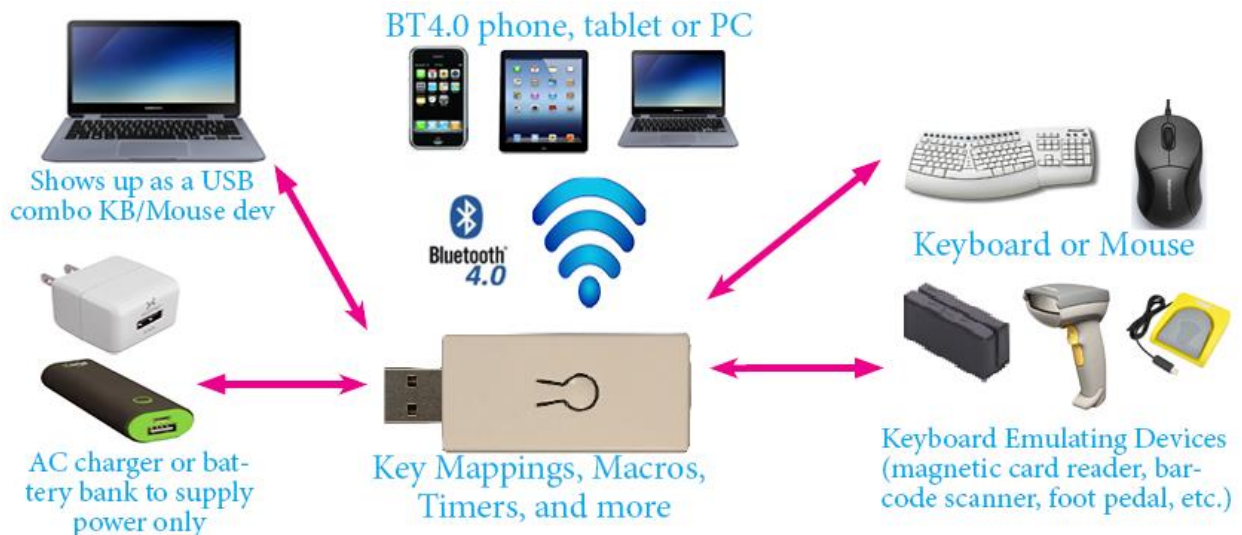
1. Introduction & Quick Start	7
2. Basic Features	9
2.1 Bluetooth 4.0 (BLE).....	9
2.2 USB Bridging	9
2.3 Keyboard/Mouse Compatibility	9
2.4 LED Indicators	10
2.5 Mode Button	10
2.6 Enable/Disable and Switch between Bluetooth/USB	11
2.7 Restore to Factory Defaults	11
3. Key Names	12
3.1 Basic Key Names.....	12
3.2 Compound Key Names.....	12
3.3 Generic Representation of Key Names.....	12
3.4 Mouse Clicks and Movements.....	13
4. Command Mode and List of commands	14
4.1 Entering and Exiting Command Mode	14
4.2 List of Commands.....	14
5. Key Mapping.....	18
5.1 Simple Mappings	18
5.2 Advanced Mappings	19
5.3 Key Mapping Command	19
5.4 Key Mapping Examples	20
6. Advanced Features.....	22
6.1 Macros	22
6.2 Timers.....	23
6.3 Output Throttling	23
6.4 Layer.....	24
6.5 Repeats	25
7. DIY Wiring and Considerations	26
8. Troubleshooting and Support.....	27
9. FCC/CE Certification.....	28

1. Introduction & Quick Start

Thank you for purchasing our Bluetooth Adapter for Keyboard and Mouse. An illustration of the BT-500 is shown below:



The following picture shows the typical use of the Adapter.



The following quick-start explains how to set up and use the adapter without going through the whole User Manual.

1. Plug the Adapter into a USB power source which can be a battery power bank, a wall power supply or a USB port on a computer. The blue LED should start blinking.

2. Plug the keyboard into the Adapter. When the keys are pressed, the green LED should blink. If not, try plug the keyboard into the adapter first, then plug the adapter into the power source.
3. To use the Adapter with a host (i.e., smartphone, tablet, PC, Mac) via Bluetooth, turn on Bluetooth on the host and search for Bluetooth devices. The Adapter should show up as “*adapter_nnnn*” where *nnnn* is the last 4 digits of its MAC address. Note the host must support Bluetooth 4.x. (Windows 7 does not have that support. Only 8 and later do).

Note: By default, if both the Bluetooth and USB interfaces are enabled (factory default), only the Bluetooth interface is active. If you would like to use the USB interface, you will need to use one of the shortcuts to switch between the two. The shortcut ctrl+alt+b switches to Bluetooth and ctrl+alt+u to USB.

The Adapter will be in discoverable mode for 30 seconds. After that you will need to power cycle it for it to become discoverable again for another 30 seconds.

4. Click on the Adapter name. There is no need to enter a pairing code. Click Yes if any confirmation dialog shows up.
5. Wait for a second or two. The status of the Adapter should change to “connected”. Now you can start using the keyboard wirelessly.

You only need to do the pairing process once. Next time the Adapter starts up, it will look for a paired host to connect. If there are multiple of them, it will connect to the first one it sees. Once the Adapter is connected, it will not be discoverable. The connection is secure per Bluetooth specification.

6. To switch between USB and Bluetooth, short-press the mode button or use one of the shortcuts: ctrl+alt+u to switch to USB only; ctrl+alt+b to switch to Bluetooth only and shortcut ctrl+alt+pause to cycle through 3 states: USB only, Bluetooth only and both active.
7. To enter command mode, press and hold the mode button for more than 2 seconds, or use default shortcut ctrl+alt+ins.

The rest of this manual explains all the features in details.

2. Basic Features

This section explains the features of the Adapter.

2.1 Bluetooth 4.0 (BLE)

- Based on Bluetooth 4.0 (Bluetooth Low Energy or BLE) with reduction in power consumption and fast connection time.
- Class 2 Bluetooth radio (2.5mw/4dBm) with distance of 10 meters.
- Compatible with all types of hosts including MacOS, iOS (iPad & iPhone), Android tablets/smartphones, newer versions of Windows and Linux as long as they have Bluetooth 4.0. (Win 7 does not support BT 4.0. Only Win 8 and up do).
- No code pairing: the adapter will pair with a host without using a keycode. Security precaution must be taken to make sure the adapter is paired with the intended host.

Currently no-code pairing is the only option. If security is your concern, we suggest disabling Bluetooth when it is not used, or always connected to a host.

2.2 USB Bridging

In addition to supplying power, the Adapter's male USB port is a functional composite keyboard/mouse device. When plugged into a host supporting USB HID functionality (such as a PC, Mac, Android with USB host cable), it shows up as a keyboard and a mouse. The keystrokes are "passed through" to the host, as if the keyboard was directly connected to it.

There are limitations in this passing through:

- The Adapter only supports 6-key rollover, even if the keyboard supports n-key rollover for $n > 6$.
- The resolution, or DPI, of a connected mouse may be different from direct connection.
- The USB device IDs (vendor ID and product ID) are different. Therefore, any software from the keyboard vendor relying on the IDs will not work.

2.3 Keyboard/Mouse Compatibility

The Adapter should work with most keyboards and mice in the market. We have tested more than 250 of them. Our compatibility page (<http://handheldsci.com/comp>) lists the high-end mechanical keyboards (most of them in the market) we have tested.

- Compatible with most USB keyboards including those with a built-in hub (e.g., Happy Hacking v1, DAS, Apple, Dell, IBM/Lenovo) and keyboard/trackball combos (both the keyboard and trackball will work).
- Most media keys on the keyboards will work including Volume +/-, Mute, Play/Stop, Previous/Next, Browser, Calculator, Home, Help, Search and Sleep. The handling of those keys varies greatly by the host. Programmable keys (for example, those marked as G1, G2, ... on some Logitech keyboards) will work only if they are programmed as regular keys.

- Many keyboard-emulating USB devices will also work if they appear as a keyboard in the operating system. Those include keypads, foot pedals, bar code scanners, magnetic strip readers, and PS2 to USB converters.
- Most 2.4G wireless keyboards will also work. We have tested those from HP, Microsoft (All-in-One, Sculpt), Logitech (Unifying Adapter), etc. However, using a wireless keyboard with the Adapter may experience increased latency since the signal has to traverse two wireless links in the same frequency band.
- The firmware has the support for a USB hub with up to two devices attached. It should work with either a built-in hub in a keyboard or an external hub. Keyboards with a built-in hub are supported provided there is no other devices connected to the hub. The following two setups are not officially supported, since there are too many combinations of those and we can't test them all. But there is a good chance they will work:
 - A keyboard with a built-in hub and with one additional keyboard or mouse connected to the hub
 - An external hub with one or two devices (keyboard or mouse) connected.

Currently the Adapter works only with keyboards and mice but not any other types of USB devices such as serial port adapters, game controllers and touch panels.

2.4 LED Indicators

There are two LED indicators: blue LED and green LED.

Blue LED is for Bluetooth interface state and connection status:

- Off if the Bluetooth interface is disabled.
- Blinks normal (once per second) when the adapter is not connected to a host.
- Blinks slowly (once per 2 seconds) when the adapter is connected to a host.
- Blinks fast (twice per second) when the adapter is in command mode.

Green LED is for USB interface state and keyboard/mouse activities. If the USB interface is enabled, this LED blinks once every 2 seconds. In addition, it blinks when there are keystrokes or mouse movements. This is useful in diagnosing device compatibility issues.

You can change the intensity of the LEDs, including turning them off, in command mode.

2.5 Mode Button

The mode button is overloaded with 3 functions, depending on how long it is pressed and held.

- Short press: can be set up to perform one of the two actions: switch between Bluetooth and USB interfaces (see the next section below for the exact behavior), or to send out a key or a macro (see `set button` command).
- Pressed and held for 2 seconds: enter command mode.
- Pressed and held for 10 seconds: factory reset.

2.6 Enable/Disable and Switch between Bluetooth/USB

The two interfaces, Bluetooth and USB, can be enabled/disabled individually in settings via command mode. They can also be activated/de-activated using the mode button or shortcuts.

An interface can be enabled/disabled in settings using the “set bt enable|disable” or “set usb enable|disable” command. If an interface is disabled, it will not be active unless it is enabled again. The shortcuts to switch interfaces will have no effect. So if you don't plan to use one of the interfaces, you should disable it. The factory default setting is both interfaces enabled.

When the Adapter powers up, only one interface will be active. If only one interface is enabled, then that is the one active, and always active. The short press of the mode button (or the shortcut) will not do anything, since de-activating the only enabled interface would render the Adapter useless.

When both interfaces are enabled in settings (factory default), only the Bluetooth interface will be active on start-up. However, you can use the shortcut `ctrl+alt+b` to switch to Bluetooth, or `ctrl+alt+u` to USB. In the rare case where you would like both interface active (so keystrokes will be sent to Bluetooth and USB simultaneously), you can use the `ctrl+alt+pause` shortcut, or the mode button, to cycle 3 states: Bluetooth only, USB only and both active. It is worth noting that this is also configurable. See the command “set iface_mode” for details.

The activation state will not persist upon power cycle or restart.

2.7 Restore to Factory Defaults

If you wish to restore the Adapter to its factory default, press and hold the mode button for more than 10 seconds. It will enter command mode after 2 seconds. Ignore that and continue to hold. The device will eventually turn itself off, and factory reset is executed.

A factory reset performs the following:

- All settings in layer 1 is reset to default values: all mappings/macros/timers will be erased. Shortcuts are back to default. Inactivity timeout is disabled.
- Initial layer is set to layer 1
- Settings in all other layers will be *NOT* be changed.

3. Key Names

Many commands take key names as parameters so it is important to understand them. Key names are used to find the corresponding keycode. In USB technology, each key name or key code represents one key in the keyboard. Although some keys have two symbols associated with them (e.g., “1” and “!” are the same key), there is only one unique keycode associated with that key.

The “+” character is used to indicate multiple keys (compound keys) so it is not used in key names. Other than that, any ASCII characters can be used in key names.

There is no separate keycode for “upper case” or “lower case” like in ASCII character set. To generate an uppercase character, one needs to emulate the actual typing. For example, to generate upper case character A, one needs to do “shift a”. This can be accomplished by a macro.

3.1 Basic Key Names

A basic key name represents just one single key. There are three groups of those names: modifier keys, media keys and normal keys.

- *Modifier keys*: there are eight of them: `l_shift`, `l_ctrl`, `l_com`, `l_alt`, `r_shift`, `r_ctrl`, `r_com` and `r_alt`.

The names `ctrl`, `com`, `alt` and `shift` are not basic key names. They are ambiguous since there are left and right of those. Nevertheless, those names are accepted in some situations, such as in the mapping destinations, as abbreviation to indicate either left or right modifier keys.

- *Media keys*: `vol_inc`, `vol_dec`, `mute`, `stop`, `play`, `mute`, etc.
- *Normal keys*: any key except the above are normal keys. Those include a-z, 0-9, f1-f12, all keypad keys, and so on.

You can list all basic key names supported in the firmware by using the “`key_name`” command.

The three lock keys, namely Caps lock, Num lock and Scroll lock, are not currently supported in any functions. So the key names do not include them.

Some keyboards, such as those from Logitech, have programmable keys marked as G1, G2, etc. If they are programmed as keyboard keys, then they are just like any other normal keys. If they are custom keys that rely on software or device driver from the vendor, then they are not supported since they don't generate standard USB HID keycodes.

3.2 Compound Key Names

A compound key name is one or two modifier keys plus a basic key, such as `ctrl+a` or `ctrl+alt+ins`. Note there is no space between the names. A compound key name can be used in some commands where a key is expected. They are also used in shortcuts (command mode shortcut, interface activation shortcuts and layer switching shortcut).

3.3 Generic Representation of Key Names

In all the places where a basic or compound name is expected, you can also use the generic form of names. A generic name is a 4-byte hex number starting with prefix `0x`. The first byte is

the Usage Page, the second byte is always 0 (reserved). The third and fourth bytes is the keycode (or Usage in USB terminology). For instance,

```
0x7000004  'a'                // regular keys are in Usage Page 0x7
0x7000005  'b'
0x70000e0  'l_ctrl'
0x70000e1  'l_shift'
0x70000e2  'l_alt'
0x70000e3  'l_com'
0x70000e4  'r_ctrl'
0x70000e5  'r_shift'
0xc0000e9  'volume increase' // media keys are in Usage Page 0xC
0xc000192  calculator
0xc000223  AC home
```

Using the generic form, you can come up with any key output, even those whose names are not in the name list, or those not on a keyboard. This is a very powerful and useful feature.

The keycodes can be found in the document *USB HID Usage Tables v1.12* from usb.org.

3.4 Mouse Clicks and Movements

The Adapter can also output mouse actions: movements and button clicks. They can be used in most places where a key name is expected. For example, you can map a simple key or compound key to some mouse movements. Mouse actions can be specified with one of the two forms:

```
mouse:{button}:{x_dir}:{y_dir}
mouse_ex:{wheel}:{pan}:{z_dir}
```

The first form specifies the button click, horizontal and vertical movements. The mouse click is a bit map so button 1 is 1, button 2 is 2, button 3 is 4, buttons 1 & 2 at the same time is 3, and so on. For horizontal direction, moving left is negative and right positive. For vertical direction, up is negative and down positive. The unit is pixel (or however the host interprets the numbers). The range of the values is -126 to 126. For example,

```
mouse:1:0:0 – click the first (left) mouse button.
mouse:0:5:0 – move mouse right 5 pixels.
mouse:0:0:-10 – move mouse up 10 pixels.
```

The second form specifies the wheel, pan and z direction movement. It is used less often. The interpretation of those in the host is not standard. It also depends on what the current application is. For example, if the current application is a web browser, the wheel action would cause the page to scroll. In a CAD application, the pan action usually causes it to zoom. Here are some examples:

```
mouse_ex:50:0:0 – move mouse wheel down (scroll down) 50 units.
mouse_ex:0:-5:0 – pan (zoom) -5 units.
```

4. Command Mode and List of commands

4.1 Entering and Exiting Command Mode

Command mode is used to configure adapter features. In this mode, the keystrokes from the keyboard are intercepted and interpreted by the Adapter as commands. The responses to the commands are sent to the host, via any active interfaces (Bluetooth and/or USB).

This is the biggest innovation in this product. With other products, such as those so-called programmable keyboards, a user needs to either rebuild the firmware, or install some kind of configuration software. Either way it is very involved. Setting up the build environment requires a lot of work and expertise. Maintaining a configuration software isn't easy, given the number of operating systems and evolving environment. Our approach is truly platform independent. It works with any operating system, without installing additional software. Moreover, the feature is in the Adapter itself. As long as you have the Adapter, you can use the feature, independent of anything else.

To enter command mode, plug the adapter in a host or pair it with a host. Open a text app (i.e., Notes, Notepad, Word, or anything that can display text). Make sure when you type on the keyboard connected to the adapter, the keystrokes appear on the app. Press and hold the mode button until the following text is emitted (which will happen in 2 seconds):

```
Bluetooth Adapter Firmware v5.4.5
Copyright (C) 2021 Handheld Scientific, Inc.
Type help or ?
L1>
```

Alternatively, you can use the shortcut to enter command mode, so you don't need to reach to the adapter to press the button. The default shortcut is ctrl+alt+ins. That 3-key combination is pretty much like the well-known ctrl-alt-del 3-finger solute in Windows. But instead of the Delete key, the Insert (ins) key is used. In the unlikely event that this 3-key combination is used by some other application, you can change to use a different key combination, or even disabled the shortcut. See "set com_mode_shortcut" in the command list.

If you see other non-English characters instead of the above text, make sure the language setting of the operating system is English. You may also need to change the keyboard layout to English since the commands are in English. Once in command mode, features such as timers and key mappings are disabled. In other words, you need to exit command mode to see those features working, if they are set up.

To exit command mode, press the mode button again or type the "exit" command.

The command prompt, "L1>", indicates which layer it is currently on. Please see "layer" in the Advanced Features section. The firmware version in the above text may be different since we continue making improvements and releasing new firmware.

4.2 List of Commands

Commands are very similar to those in Unix Shell where tokens are separated by space and terminated by <Return> or <Enter>. All commands are in lower case. In the following command list, curly brackets {} signifies a token that needs to be substituted by the actual value. For

example, {time} should be 10 or whatever value you want to set. Square brackets [] signifies the parameter is optional. Vertical bar signifies “or”. For example, enable | disable means either enable or disable as the argument.

Once in command mode, you can start with typing “help” or “?” for a list of commands, and “help {command}” for help with a specific command. Our goal is to make the help messages concise and comprehensive so one is able to understand and use the commands which consulting this manual.

It is important to note that the changes will be lost upon reboot or power cycle unless the “save” command is issued. This command saves all settings on the current layer to persistent storage. Also the current layer when the Save command is executed is saved as the start-up layer, whose settings will be loaded in the next start-up.

The following is a complete list of commands in this firmware version.

`syntax`

Display concise description of command syntax.

`show`

Display useful adapter information such as the ID of attached keyboard, LED intensity, throttle interval, interface status and shortcuts.

`save`

Save changes so they will persist across re-starts. If you don’t execute this command, all changes will be lost when the adapter is re-started or power cycled.

`exit`

Exit command mode. Same as pressing the mode button. Together with the shortcut to enter command mode you don’t need to reach out to the adapter to switch modes.

`device clear`

Clear paired hosts in the adapter. Even after this command is executed, a paired host can still connect to the Adapter. So if you wish to pair the Adapter with a different host, you will also need to unpair it from the current paired host.

`reboot [hard]`

Restart adapter so you don’t need to unplug it from power. Optional argument “hard” to perform a factory reset, the same effect as press and hold the mode button for 10 seconds.

`key_name`

Print out the list of key names. See the Key Name section in this manual for more details This is displayed in multiple pages. Type the same command to go to the next page.

set

Set system parameters. There are multiple sub-commands.

```
set usb enabled|disabled
```

Enable or disable the USB interface. Default is enabled.

```
set bt enabled|disabled
```

Enable or disable the Bluetooth interface. Default is enabled.

```
set bt name {name}
```

Set the device name to name. Example: `set bt name my_keyboard`. No space is allowed in the name. You may need to unpair then re-pair the Adapter for the host to display the new name.

```
set led blue|green {n}
```

n=0-100. Set LED intensity. Set it to 0 to turn off the LED. Note the intensity is not linearly proportional to the value. For example, the value may be down to 20 for the intensity to be reduced by half.

```
set time throttle {n}
```

set output throttling time to n milli seconds (ms). See Output Throttling in Advanced Feature section.

```
set com_mode_shortcut enabled disabled|{compound_key}
```

Enable/disable shortcut to enter command mode, or set the shortcut key. The default is enabled with key `ctrl+alt+ins`. Examples:

```
set com_mode_shortcut disabled – disable command mode shortcut
```

```
set com_mode_shortcut ctrl+alt+c – change the shortcut to the one specified.
```

```
set iface_modes 2|3
```

Set how the `iface_shortcut` cycles the modes. If set to 2, the shortcut cycles through 2 modes: USB active only and BT active only. If set to 3, the shortcut cycles through 3 modes: USB active only, BT active only and both active.

```
set iface_modes 2 – set the iface_shortcut to cycle between 2 modes.
```

```
set iface_shortcut enabled disabled|{compound_key}
```

Enable/disable the shortcut to switch Bluetooth/USB interfaces. The default is enabled with key `ctrl+alt+pause`. Examples:

```
set iface_shortcut disabled – disable this shortcut
```

```
set iface_shortcut ctrl+alt+f12 – change the shortcut to the one specified
```

```
set bt_shortcut enabled disabled|{compound_key}
```

Enable/disable the shortcut to switch to Bluetooth interface. The default is enabled with key `ctrl+alt+b`. Examples:

```
set bt_shortcut disabled – disable this shortcut
```


`set bt_shortcut ctrl+alt+t` – change the shortcut to the one specified

`set usb_shortcut enabled disabled|{compound_key}`

Enable/disable the shortcut to switch to USB interface. The default is enabled with key `ctrl+alt+u`. Examples:

`set usb_shortcut disabled` – disable this shortcut

`set usb_shortcut ctrl+alt+s` – change the shortcut to the one specified

`set button mode iface|key`

Set the mode button to either switch between BT and USB interfaces, or to send out a compound key.

`set button key {compound_key}`

Set the compound key to be sent out when the mode button is short pressed. The mode of the button must be set to key. Examples:

`set button key a` – set the key to a. i.e., short press of the mode button will send key 'a'

`set button key macro:1` – short press of the mode button will send macro 1.

`timer` – timer related command. See “timer” in Advanced Features section.

`map` – key mapping related command. See the “Key Mapping” section in this manual.

`macro` – macro related command. See “macro” in Advanced Features section.

`layer` – layer related command. See “Layer” in Advanced Features section.

5. Key Mapping

Key mapping is such an important feature that it deserves its own section in this manual. With key mapping, every keyboard becomes programmable. The possibility is unlimited. For instance, you can set up a keypad as a media control center by mapping the keys to volume up, down, stop, play, etc.

It is worth noting that the Adapter intercepts the USB reports from the keyboard to perform the mapping. Therefore, it has some limitations as specified below. It can't do arbitrary mapping like the firmware in a keyboard can but the limitations have little impact on the usefulness of this feature.

In a mapping, a map source is translated to a destination. When the map source key or key combination is entered, the destination key or key combination is generated. The map source key is "consumed" and will not be further processed. Based on their usage in key mapping, the keys on a keyboard can be divided into 4 categories:

- Regular keys. Those are the keys like a-z, 0-9, F1-F12, keypad keys, etc.
- Modifier keys. There are 8 modifier keys: left shift (`l_shift`), left control (`l_ctrl`), left alt (`l_alt`), left com (`l_com`) and their right counterparts (`r_shift`, `r_ctrl`, `r_alt` and `r_com`). The com keys are also called the win keys.
- Media keys: media keys are those like volume increase/decrease, pause, previous/next, browser, calculator, etc. Those keys are in the USB HID Usage Page 0xC. Currently media keys can only be in map destinations.
- Three lock keys: caps lock, num lock and scroll lock. Those usually have a LED associated with them. Due to their specialty, those keys can't be in map source or destination. They are excluded from key mappings.

Depending on the map source and destination there are 3 types of mappings:

1. A simple regular key is mapped to a regular key, a media key, a compound key or a macro.
2. A modifier key is mapped to another modifier key.
3. A compound key is mapped to a regular key, a media key, another compound key or a macro.

Other types of mapping are not supported at this moment. For instance, you can only map a modifier key to another modifier key but not any other types of keys. Media key cannot be used as map source.

The first two types of mappings above are called Simple Mappings and the third one Advanced Mappings. They are discussed in detail in the next sections.

5.1 Simple Mappings

A simple mapping is one where the source key is a single key, either a simple key or a modifier key. Simple mappings are implemented by efficient memory mapping algorithm so there is little performance impact. As a result, you can set up as many as you would like without performance penalty.

When mapping a modifier key, you will need to specify whether it is left key or right key. For instance, you need to use either `l_ctrl` or `r_ctrl` instead of just `ctrl` which is ambiguous.

5.2 Advanced Mappings

Advanced mappings are the ones where the source key is a compound key such as alt+f1 or ctrl+alt+s. Since they have to be searched and matched for each key stroke, there could be performance penalty. Therefore, the number of advanced mappings is limited to 16 and they should be used judiciously. We will see some examples of each types shortly.

5.3 Key Mapping Command

Key mapping is managed by the map command. Its usage is explained in the following session.

```
map [list]
```

The map command, without any parameter, lists the mappings in two groups: basic mappings and advanced mappings.

```
map add {map_source} {map_destination}
```

Add a key mapping. map_source can be a simple key (basic mapping) or a compound key (advanced mapping). If a mapping with the same map_source, it will be replaced. See mapping examples in the next section for how to use this command.

```
map del {n}|all
```

Delete the nth mapping. n is the index number when you display all the mappings. You can also use del all to delete all mappings, where you will be prompted for confirmation.

```
map add dvorak|colemak
```

Add the mappings to turn a keyboard with Qwerty layout to Dvorak or Colemak layout. You can use the “map” command to see what mappings are added. You can do this in a layer and switch layers using a shortcut to quickly change the layout.

```
map names
```

Display all the basic key names. This is the same as the key_name command.

```
map example
```

Display some mapping examples. If you are reading this document, there are a lot more examples in the next section than in the online help.

5.4 Key Mapping Examples

This section shows key mapping examples. Keep in mind that you need to execute the “save” command for any changes to survive re-starts.

```
// swap the left alt and com keys, to use a Windows keyboard with a Mac, or vice versa
map add l_alt l_com
map add l_com l_alt

// IMB Model-M and Northgate OmniKey keyboards don't have a com key
// So remap right alt to com
map add r_alt r_com          // turn the right alt key to com key

// some simple examples
map add a b                  // map 'a' key to 'b'. Just a test. Not really useful
map add f1 macro:0          // f1 key to macro 0
map add ctrl+alt+f3 macro:1 // advanced mapping example. Limited to 16 of those.

// Turn the keypad keys to media controls. Now there is a reason for a full-size keyboard.
map add kp_minus vol_dec    // keypad minus to reduce volume
map add kp_plus vol_inc    // keypad plus to increase volume
map add kp_enter play      // keypad enter to start/stop playing
map add kp_0 prev          // keypad 0 to previous track
map add kp_del next        // keypad del to next track
map add kp_7 mute          // all other keys to mute, so you can stop the music
map add kp_8 mute          // quickly when the boss walks in 😊.
map add kp_9 mute          //

// No keypad? No problem. Use function keys.
// Note those are advanced mappings with the number limited to 16.
map add alt+f7 vol_dec     // alt+f7 to reduce volume
map add alt+f8 vol_inc     // alt+f8 to increase volume
map add alt+f9 mute        // alt+f9 to mute
map add alt+f10 play       // alt+f10 to start/stop playing
map add alt+f11 prev       // alt+f11 for previous track
map add alt+f12 next       // alt+f12 for next track

// What? Your keyboard doesn't even have function keys? We got you covered.
map add alt+- vol_dec      // alt+- to reduce volume (the minus key)
map add alt+= vol_inc      // alt+= to increase volume (the plus key)
map add alt+delete mute    // alt+delete (backspace) to mute
map add alt+, prev         // alt+, (the < key) previous track
map add alt+. next         // alt+. (the > key) next track
map add alt+/? play        // alt+/? (the ? key) start/stop

// Use generic key name. See section for key names. You can send anything defined by the
// standard. How the host interprets those varies greatly.
map add alt+f2 0xc000192   // 0xc000192 is the keycode for "Calculator". This will bring
                           // up the calculator in Windows
```

```
map add alt+f2 0xc000223 // 0xc000223 is the keycode for "Home". This will usually
                          bring up a Browser in Windows (OS dependent)

// control mouse actions using a keyboard. You can set up a macro to do a sequence of
// those actions.
map add alt+j mouse:0:-5:0 // move mouse left 5 units
map add alt+i mouse:0:0:-5 // move mouse up 5 units
map add alt+k mouse:0:0:5 // move mouse down 5 units
map add alt+l mouse:0:5:0 // move mouse right 5 units

map add alt+r_arrow mouse:2:0:0 // brings up mouse right click menu
map add alt+up_arrow mouse_ex:1:0:0 // If the current focus is browser, scroll up 1
                                     unit.
```

Again, you need to use the "save" command for the changes to stick across reboots.

It is worth noting that those are all done in one layer. You can set up different mappings in different layers, and switch between them quickly using the layer switching shortcut. See "Layer" in the "Advanced Features" section.

6. Advanced Features

This section describes other useful features such as macros, timers and layers.

6.1 Macros

A macro is a series of keystrokes or mouse movements (or mix of them). It can be used in many places where key names are expected. For example, you can map a basic key or a compound key to a macro. Or set up a timer to send out a macro periodically. The way to specify a macro as a key name is `macro:n` where `n` is the n^{th} macro.

There is a special token, `delay`, that can be used in macros. The format is `delay:n` where `n` is in milli seconds. The effect of this token is insert a delay of `n` milliseconds in the processing.

There can be up to 64 macros in the adapter. The number is from 0 to 63, inclusive. Macros cannot be nested. The command to manage macros is `macro`.

`macro` – the command with no argument lists all macros.

`macro edit {n}` – edit the n^{th} macro where `n=0-63`.

Once the edit command is issued, the input is expecting a series of key names separated by space, terminated by `<Enter>` or `<Return>`. You can use any basic key names, and the delay token. The modifier key names `shift`, `ctrl`, `alt` and `com` can be used, or their left or right representations.

The following examples show how to setup some macros:

Output	Text to input, terminated by <code><Enter></code> or <code><Return></code>
hello	h e l l o
Hello<enter>	shift h shift e l l o enter
hello<delay 1s>world	h e l l o delay:1000 w o r l d
(increase volume 5 times)	vol_inc vol_inc vol_inc vol_inc vol_inc
(move mouse 5 units right, then click left button)	mouse:0:5:0 mouse:1:0:0

There is no command to delete macros, since macros do not have any effect unless they are used. If you don't want to use them, remove them from where they are used (in mappings and timers) instead of deleting the macros themselves.

6.2 Timers

Timers are used to send out keystrokes or mouse movements periodically. They are useful in preventing connections (wired or wireless) from timing out, or the host from entering into sleep due to inactivity, hence this feature was previously called “keep alive”. You can setup up to 4 timers.

Output from timers is sent to all interfaces that are enabled and active.

By default, all timers are disabled. Before using a timer, you will need to set a period and a key by using the `timer` command. The key can be a basic key, a compound key or a macro. Period is in seconds. The period of 0 disables the timer. The following describes the command.

`timer` – lists all 4 timers with their periods and keys.

`timer {n} period {m}` – set the period of timer n (n=1-4) to m seconds. Setting period to 0 disables the timer.

`timer {n} key {key}` – set the key of timer n to key which can be a basic key, a compound key or a macro.

```
Examples:  timer 1 period 60      // set timer 1 to send keys every 60 seconds
           timer 1 key a          // set timer 1 to send the key 'a'
           timer 2 key macro:0    // set timer 2 to send macro 0
```

6.3 Output Throttling

Many USB input devices such as barcode scanners and magnetic card readers emulate keyboard for plug and play. As such they can be used with the Adapter. Those devices are designed to communicate via USB with fast data rate. Moreover, they send out relatively large amount of data. For instance, a magnetic card reader (credit card reader) may send out more than one hundred characters on one swipe.

On the other hand, the Bluetooth data rate is much lower. This data rate also varies greatly depending on the quality of the Bluetooth chip in the hosts and how busy the shared air channel is. As a result, the fast USB output may cause loss of data when transferred via Bluetooth.

In anticipating this problem, the Adapter implements an output throttling feature where a delay is intentionally inserted between each character. The command to enable output throttling is:

```
set time throttle {time}
```

where {time} is the delay in milli seconds (ms). If time is zero, this feature is disabled (no delay is inserted). Examples:

```
set time throttle 10 // set throttling delay to 10 ms
set time throttle 25 // set throttling delay to 25 ms
```

Throttling is only applied to the Bluetooth interface. It does not affect the USB interface. When you increase the throttling time, you will see the output slowing down. Test with your setup to make sure all the data is reliably received by the host.

6.4 Layer

A layer is a group of settings (mappings, macros, timers, etc.) that can be switched together easily. It provides a way to completely change the layout or functions of a keyboard with a simple shortcut. There are 4 layers (1-4) in this version of the firmware. When you are using the keyboard or in command mode, you are always working on a specific layer. All settings you create are only pertinent to that layer. In command mode, the input prompt, “Ln>”, where n=1-4, tells you which layer you are currently on.

There are two ways to switch to a different layer: (1) In command mode, the command “layer n” switches to the nth layer; (2) In normal mode, the layer switching shortcut (default is ctrl+alt+l, where the last character is lower case L) will cycle the layers (from 1 to 2 to 3 to 4 and back to 1). When switching layer using the shortcut, a prompt of 4 characters “ Ln “ (space, letter L, number n, space) will be sent out to the host, as if they were typed from the keyboard. This is to inform the user which layer is currently on. You can enable/disable the prompting in settings if you don’t find it useful.

When you switch to a different layer, the saved settings of that layer are loaded, and all unsaved changes will be lost. Therefore, if you make any changes in settings, save them before switching layer.

When the “save” command is issued, all settings will be saved on the current layer, as well as the layer number itself. Next time the Adapter powers up, it will start on that layer, and load all the settings of that layer.

If a factory reset is performed (either by holding the mode button for more than 10 seconds or by the command “reboot hard”), only the settings in the first layer are reverted to defaults, and the startup layer is set to 1. Settings in other layers will not be affected.

The command to use with layer is “layer”. The following is its usage:

```
layer – without an argument, this command prints the information related to layer.  
layer n – switch to layer n, where n is 1, 2 3 or 4.  
layer prompt enable|disable – enable/disable the layer switching prompt  
layer shortcut enable|disable|{compound_key} – enable/disable layer switching  
shortcut or set the shortcut
```

Examples:

```
layer 3 // switch to layer 3  
layer prompt disable // disable layer switching prompt (default is enabled)  
layer shortcut enable // enable layer switching shortcut (default)  
layer prompt alt+f12 // set layer switching shortcut to alt+f12  
layer prompt ctrl+alt+s // set layer switching shortcut to ctrl+alt+s (default is  
ctrl+alt+l)
```


6.5 Repeats

Normally, a press and release of a key generates one key output. What if you want to generate keystrokes continuously, which is useful in some of those shooting games? You have two options, either keep hitting the keys repeatedly, or press and hold the key where the operating system generates the keystrokes at a pre-determined interval. With the “repeat” feature of the adapter, now there is a better way to do that.

When a repeat is setup, a key is used like a toggle switch. When the key is pressed and released, some keystrokes are output continuously, at the specific interval, until the same key is pressed and released again. The key used to toggle the output is called the trigger key. The key that is output continuously is called the target key. Both can be compound keys. In some cases the trigger key and the target key are the same but they could very well be different. Users can set up up to 4 repeat keys.

The command to set up repeat is “repeat”, with the following usage:

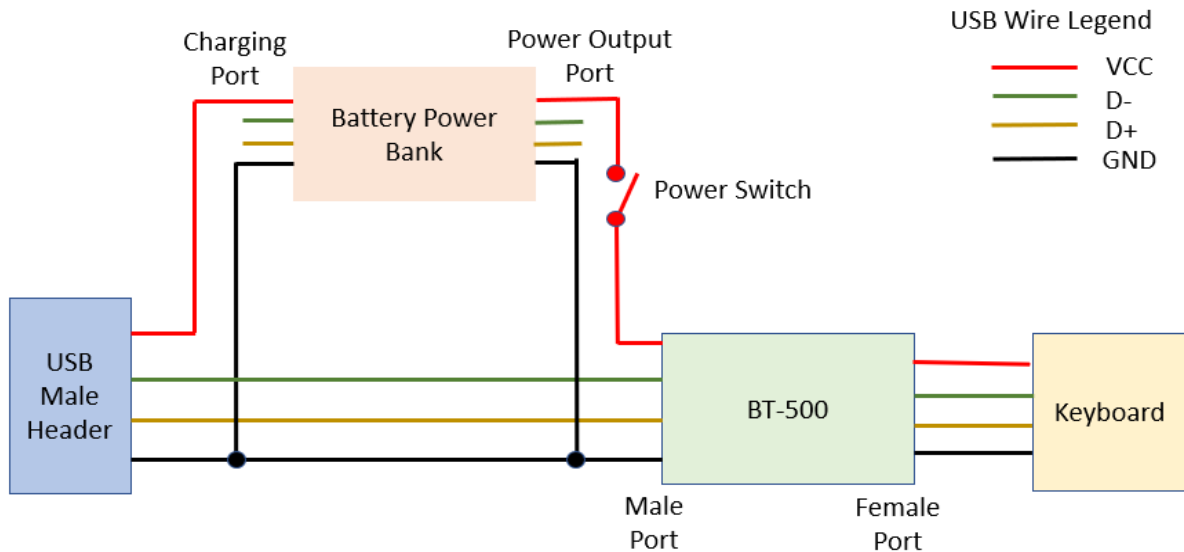
```
repeat – without an argument, this command shows the key and period for all 4 repeats.  
repeat {n} period {m} – set the period of repeat n to m milli seconds. N=1-4. M ≥ 10  
repeat {n} key {triggerKey} [{targetKey}] – set the key of repeat n to the  
compound key key
```

Examples:

```
repeat 1 period 100           // set the period of repeat #1 to 100 ms  
repeat 1 key a                // set the key of repeat #1 to 'a'. So the 'a' key will act as a  
                             // toggle. Press and release the key once, key 'a' will be  
                             // generated at 100ms, until the key is pressed and released  
                             // again.  
repeat 1 key f1 mouse:0:10:10 // the f1 key will toggle the mouse movements  
repeat 1 key f2 macro:1       // the f2 key will toggle the output of macro #1
```

7. DIY Wiring and Considerations

The small size of this device makes it ideal for DIY projects. The circuit board can be molded inside a USB keyboard with a battery power bank to make it wireless. The following shows how to wire them together.



DIY Wiring Diagram: to mold a BT-500 into a keyboard

The size of the circuit board is 66x20mm with the two connectors, or 50x20 without. The thickness is 10mm with the connectors, or 3.5mm without the connectors and the mode button.

The current consumption is about 40mA. The power from the USB male connector is passed through directly to the female connector.

If you plan to use a battery to power the board, the voltage to the circuit board can be as low as 3.6V. However, if you are modifying a USB keyboard, which usually requires 5V supply voltage, a circuit to convert (boost) the battery voltage to 5V is most likely needed.

8. Troubleshooting and Support

If the Adapter does not appear to work, please follow below those steps to troubleshoot:

1. Is the power applied? The Adapter does not have a battery. It needs power through its male connector. When power is applied, the blue LED should be blinking.
2. Plug a keyboard into the Adapter. When keys are pressed, is the green LED blinking? If not, the Adapter does not detect the keyboard or is not able to parse its descriptors. Try reversing the order: plug the keyboard into the Adapter first, then plug the Adapter into the keyboard. If green LED still does not blink when keys are pressed, contact us.
3. Is the host able to detect the Adapter when searching for Bluetooth devices? If not, this is most likely because the host does not have Bluetooth 4.0. Check its spec. Keep in mind that the Adapter can be discoverable only in the first 30 seconds of connecting power.
4. If the green LED does blink when keys are pressed, but the output do not reflect what is pressed, for example, some keys don't generate output and some keys generate the wrong output, this is probably due to the incorrect parsing of the USB report descriptor. Please contact us.

When contact us, it will be very helpful to include the following information:

- The firmware version in your unit (this is displayed when entering command mode)
- The ID of the attached keyboard (this is displayed with the “show” command)
- The brand and model of the keyboard you are using. An image of the product label, usually on the back of the product, will be very helpful.
- Describe in which step above you are having issue with.

We will seriously investigate every case of incompatibility.

9. FCC/CE Certification

BT-500

Bluetooth Adapter for Keyboard and Mouse
FCC ID: 2AQL7-BT-500

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

TCB

**GRANT OF EQUIPMENT
AUTHORIZATION**

TCB

**Certification
Issued Under the Authority of the
Federal Communications Commission
By:**

**PHOENIX TESTLAB GmbH
Koenigswinkel 10
32825 Blomberg,
Germany**

**Date of Grant: 08/09/2018
Application Dated: 08/09/2018**

**Handheld Scientific, Inc.
5201 Great America Pkwy
Suite 320
Santa Clara, CA 95054**

Attention: [REDACTED], CEO

NOT TRANSFERABLE

EQUIPMENT AUTHORIZATION is hereby issued to the named GRANTEE, and is VALID ONLY for the equipment identified hereon for use under the Commission's Rules and Regulations listed below.

FCC IDENTIFIER: 2AQL7-BT-500
Name of Grantee: Handheld Scientific, Inc.
Equipment Class: Part 15 Low Power Communication Device Transmitter
Notes: Bluetooth Adapter for Keyboard and Mouse

Grant Notes

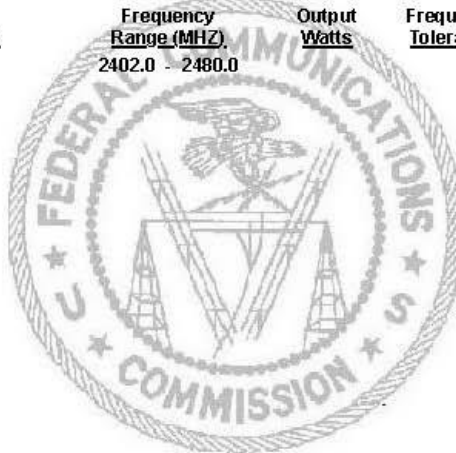
FCC Rule Parts
15C

Frequency Range (MHZ)
2402.0 - 2480.0

Output Watts

Frequency Tolerance

Emission Designator





Attestation of Global Compliance

Recognized by China National Accreditation Service for
Conformity Assessment & Taiwan Accreditation
Foundation

T: 0086-755-2908 1955
F: 0086-755-2600 8484
E-mail: agc@agc-cert.com
Web: Http://www.agc-cert.com

2F., Building 2, No.1-No.4, ChaxiSanwei
Technical Industrial Park, Gushu, Xixiang Street
Baoan District, Shenzhen, P. R. China (518000)

EU-RED Certificate of Conformity

Radio Equipment Directive (RED) 2014/53/EU

Registration No. AGC00174180603E0

Manufacturer Handheld Scientific, Inc.
5201 Great America Pkwy, Suite 320, Santa Clara, CA 95054, USA

Product Designation Bluetooth Adapter for Keyboard and Mouse

Brand Name N/A

Model / Series Models BT-500

Factory Handheld Scientific, Inc.
5201 Great America Pkwy, Suite 320, Santa Clara, CA 95054, USA

Requirement	Applied Standards	Document Evidence	Result
Art.3.1(a) Health	EN 62479:2010	Test Report: AGC00174180603EH02	Conform
Art.3.1(a) Safety	EN 60950-1:2006+A11:2009 +A1:2010+A12:2011+A2:2013	Test Report: AGC00174180603ES01	Conform
Art.3.1(b) EMC	Draft EN 301 489-1 V2.2.0 Draft EN 301 489-17 V3.2.0	Test Report: AGC00174180603EE01	Conform
Art.3.2 Radio	EN 300 328 V2.1.1	Test Report: AGC00174180603EE11	Conform



Signed by Quality Manager
Issue Date: Jul. 23, 2018



Recognized by Attestation of Global Compliance (Shenzhen) Co., Ltd., in accordance with the RED Directive 2014/53/EU. The certificate doesn't imply assessment of the production. The Applicant of the certificate is authorized to use this certificate in connection with EC declaration of conformity to the Directive. The certificate is only applicable to the equipments described above. This certificate shall not be re-produced except in full without the written approval of Attestation of Global Compliance (Shenzhen) Co., Ltd.

Note: This certificate is part of the full test report(s) and should be used in conjunction with it.

Version: 2.0

