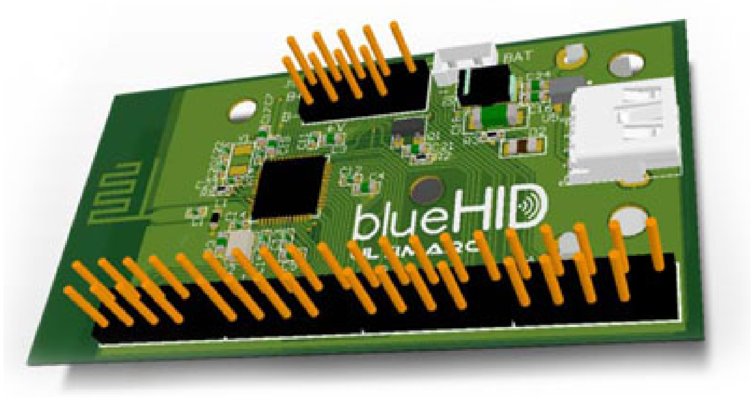
Bluetooth UNIVERSAL HUMAN INTERFACE DEVICE

TECHNICAL MANUAL



Contents

Section 1: Introduction

HID is a computing term for a class of devices called Human Interface Devices. These are certain types of PC peripherals, usually connected via USB or Bluetooth.
All modern OSes support a number of "native" HID devices using in-built drivers. The BlueHID board appears as the following HID devices on the PC:

- **Keyboard**
- **Gamepad (including analog and digital controls)**
- **Mouse**
- **LED or logic level outputs**

The way in which the board emulates natively-supported devices means it requires no special drivers or host software, with the exception of LED output control.
The board has 36 connections which can be configured “over the air” to any of these HID device types providing a truly universal Bluetooth I/O interface. For example, you can connect pushbuttons to generate keyboard keystrokes, trackballs to act as a mouse, pedals to act as analog game controller joysticks, or hundreds of other device types. The host will see all the devices as something it recognises, so all standard applications will work. The board can also be used to control LEDs or other devices.

Connections can be assigned as:

- Keyboard key
- Gamepad Button
- Gamepad POV Hat
- Gamepad analog axis
- Gamepad analog axis with auto calibration and dead zone
- Mouse button (including double-click)
- Quadrature Mouse Pair. X or Y axis. Normal or reverse direction.
- Quadrature left/right button presses.
- Host-controlled LED output using special software
- PC Num/Scroll/Caps lock LEDs
- BlueHID locally-controlled LED output based on the state of any switch.
- 5 Volt output (20ma max)
- Ground

Features

- No special drivers or application changes needed except for host-controlled LEDs which has an API with code examples.
- Simple open pin-header connectors allow use of separate or composite modular wiring harnesses.
- U-Config configuration utility allows the connections to be configured for each device type, and keycodes and other attributes set.
- Configuration information is sent “over-the-air” from a PC equipped with a standard Bluetooth interface.
- Switches handled by super-fast logic using an individual de-bounce state counter for each input giving ultimate performance for gaming applications.
- No key ghosting or blocking ever.
- Advanced keycode/button programming with in-built shift feature. Includes macro capability
- 12-bit analog axis support
- Configuration utiity interacts real-time with the board and all settings stored instantly in Flash ROM on the board.
- Optional CR2032 battery holder. Or Optional USB power connector and on-board charging circuit for external battery.
- 36 Truly independent inputs allow all switches to be commoned to ground. No matrix or diodes
- Modular 8-way + ground harnesses available.
- Extended switch harnesses available.
- ESD protected inputs. Vital for high-footfall public applications.
- All settings stored in Flash ROM and retained after power-off.
- Configuration can be changed on-the-fly using a command-line. The board can assume a completely different layout instantly.
- Simple-to-use PC Configuration application.
- Advanced power-saving design features with standby power as low as 50 nano-amps.
- Switched power pin for external devices which turns off in sleep mode.

Advanced Features:

Key and Button functions can have one primary and one secondary code assignment.
Secondary assignments are invoked by first activating a “BlueHID Shift” connection.
Any connection can be assigned as a shift.
Unique separately-programmable “button down” and “button up” events allow use

of all types of normally-closed, normally-open, or active high/low controls.
“Button down” and “button up” can be separately assigned as primary or secondary codes.
Controls can be assigned as “normal” or “pulse”.
Toggle (push on-push off) and flip-flop modes available.
Programmable analog offset and scale factor for all analog axes.
X and Y axes can be configured with auto calibration and a dead-zone.
Extended macro capability.
User-upgradeable “over-the-air” firmware for future enhancements and customizations.

Examples of control types which can be connected

- Pushbuttons
- Switch-type joysticks
- Toggle switches
- Toggle switches with integral LED (does not require a second connection for the LED)
- Momentary pushbuttons with integral LED (LED can be controlled without any host application, or can be host-controlled)
- LEDs associated with a switch. (LED can be controlled without any host application)
- LED indicators driven by a PC application.
- Other devices can be controlled either directly or via an external high-current driver
- Analog joysticks
- Incremental Rotary encoders (Spinners).
- Low-resolution rotary encoders (360 degree volume control type)
- 12-way or N-way rotary switches used as encoders (requires simple diode circuit)
- Trackballs
- Pots
- Optical steering wheels
- Pot-type steering wheels
- Pot-type throttle pedals
- Analog Voltage sources (0-5 volts).
- TTL logic signals.

Control counts which can be configured:

Total 36 connection pins.

Max 6 analog axes on 36 possible pins
Max 32 Gamepad buttons on 36 possible pins (Windows limitation)
Max 3 High-res quadrature mouse devices on 20 possible pins out of the 36.
Max 3 Quadrature mouse axes (X, Y and Z, Windows limitation)
Max 8 Low-res rotary encoder devices in button-pulse mode.
Max 36 keyboard keys on 36 possible pins.
Max 16 LEDs or outputs in any combination of PC or BlueHID controlled. (max 20mA per LED or output)
Power pin for external devices such as encoders, pots etc which turns off when in power-saving modes.

Bluetooth SIG Qualification.

This product has passed compliance testing and is fully qualified as an “end product” by Bluetooth SIG (Special Interest Group)

Details are:

QDID (Qualified Device ID): 81294.

Declaration ID: D030542

Users can refer to our QDID in their own products using this board, which is permissible by Bluetooth SIG and will not be required to undertake further SIG qualification.

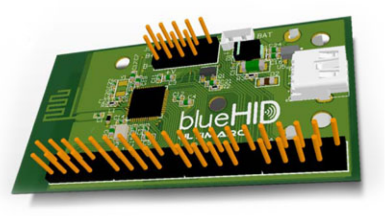
Section 2: Powering the Board and Connected Devices

The board requires power in the range of 2 volts – 5 volts. Reverse-polarity protection is built-in.

Two versions are available:

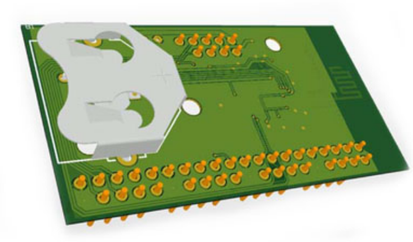
USB power with charger version

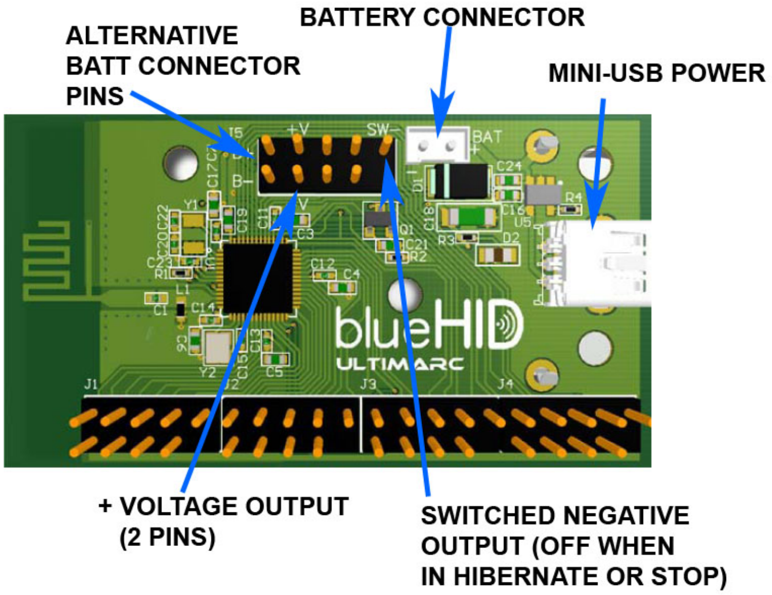
This version includes an on-board charging circuit which is powered by an on-board mini-USB connector. The USB connector is only used for power and has no data capability.
This version has a connector for plugging in a suitable lithium-ion battery.



CR2032 Battery holder version

This version is self-contained and has a battery holder on the reverse of the board. It does not have the USB connector or charging circuit.





Battery Connector

This is a 2.5mm-pitch JST B2B connector as used on many Lithium-Polymer batteries. Example shown below:

Alternative Battery Connector Pins

These pins are directly connected to the battery connector, above, and are simply an alternative way to connect a battery.

Mini-USB Power

This is a Mini-USB connector which is only used for power. No USB data functionality exists on this board. The connector can power the board and/or charge the connected Lithium-Polymer battery. Charging is rapid, average 400 milliamps of charge current.

Voltage Output Pins

These are two pins which supply power to external devices which use a relatively high current such as optical encoders.

Switched Negative Pin

This pin is used for the negative power for external devices such as encoders etc. When devices are connected, their power is switched off by means of "floating" this pin when the board goes into power-saving modes (see later for details of these modes). This pin should be used in conjunction with the high-current voltage output pins (above).

Current Consumption (excluding external devices)

Controls in normal use	4 milliamps
After 15 seconds of inactivity	0.03 milliamps
After 2 Mins of inactivity	0.02 milliamps
Sleep Mode (configurable timeout). Wake on any pin.	300 nanoamps
Sleep Mode (configurable timeout). Wake on J2-6 pin.	60 nanoamps

General Power Considerations:

Do I need a power switch?

In general if there are no external devices powered from the board there is no need for a power switch provided "Stay Connected" is not enabled in the U-Config utility.

These modes disconnect the Bluetooth link after a configurable timeout. The link is automatically re-established when a switch-type control is activated. This takes approx. 0.5 seconds (varies with different host systems).

Whether or not a power switch is required depends on the design of external devices. Although, if correctly connected, the device will be powered-down in power-saving modes, some leakage of power may still occur.

A power switch should be used if "Stay connected" is enabled and/or if permanently-powered external devices are connected.

What is the difference between the two power-saving modes?

After a configurable timeout, the board will go into one of these two modes. The mode it enters is configurable.
“Wake on any pin” mode consumes 300 nanoamps of current approx., whereas “Wake on J2-6 only” mode consumes 60 nanoamps. In this mode, power saving mode can only be exited by activity on ONE specific pin, which is J2 pin 5 (This is the Start1 button when wiring as a game controller using the U-HID Type 3 harness).

Summary of actions which “Wake” the device from power-saving modes:

“Wake on any pin”	Analog Pin	Control pushed in any direction approx. 25% (edge triggered)
	Switch Pin	Switch activated on any pin on J1 or J3 (edge triggered)
	Quadrature	Activity (see note)
“Wake on J2-6 Only”	Switch Pin	Only switch on J2 Pin 6

Note:
Although quadrature pin activity will wake the device, many quadrature devices such as trackballs will be powered down at this time so will not be able to generate any change on the signal pins.

Battery Monitoring

The BlueHID board reports battery level to the host system by the Bluetooth standard method, and can be checked on the host (method depends on host system). NOTE: Battery monitoring requires one spare pin on the board for internal use (any pin) and will not function if all pins are configured.

Power Saving Tips

- Always connect external devices as shown in this manual, to pins defined as 5V or to the “switched ground” pin, so they are powered off when in power-saving modes and power is pulsed as required during normal operation mode.
- Turn off the self-test LED in U-Config after installation and testing
- Do not configure “Stay Connected” in U-Config.

- For ultimate battery life, configure “Wake J2-6 Only” rather than “Wake on Any Pin” Mode in U-Config.

- Do not leave U-Config running as this inhibits power-saving modes.

- Do not configure any pins which are not used by controls. This is especially important with analog pins. Defining an unconnected pin as analog will pick up electrical noise and cause activity which will prevent any power-saving modes from being entered.

Section 3: Bluetooth Host Connectivity

Tested Host Systems

The BlueHID has been tested with the following hosts:

PC Windows 7, using the CSR 4.0 Bluetooth dongle with CSR PC software supplied with the dongle.

PC Windows 8 and 10, using the CSR 4.0 Bluetooth dongle *Do not install supplied CSR software* as Windows 8 and 10 have Bluetooth 4 support.

IOS All recent versions

Android All recent versions

MacOS All software and hardware which supports Bluetooth 4

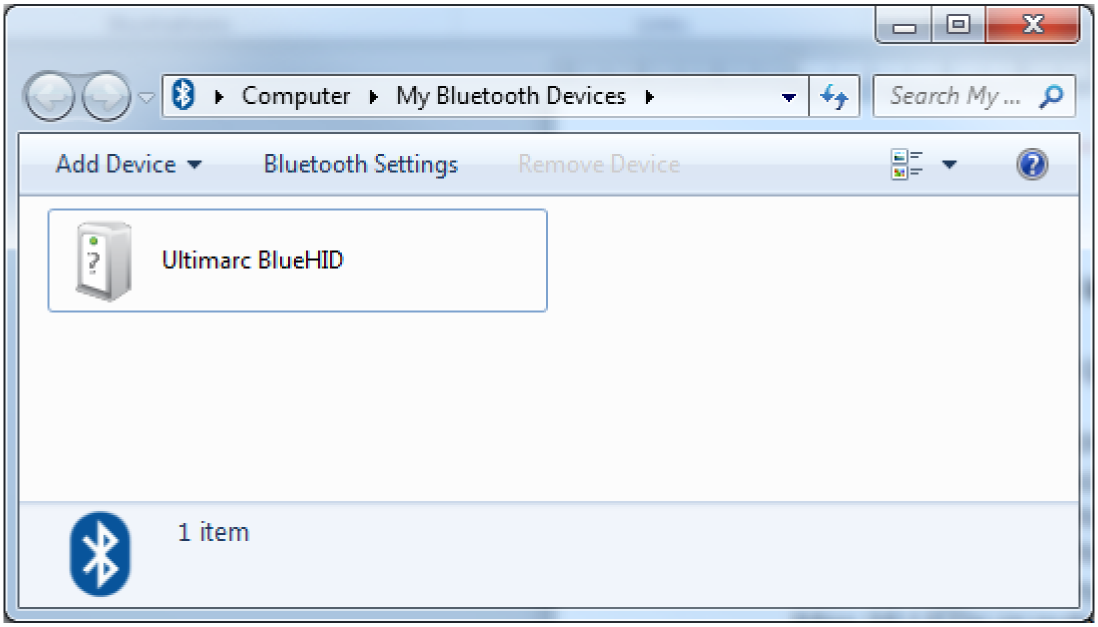
Other OSes should operate but have not yet been tested at time of writing.

Connecting to the Host

When the BlueHID is powered up, or when any control programmed as a switch is activated, the board will start to advertise its presence to the host system. During this time, the host needs to be told to search for devices. It should discover the device and pair with it. A temporary key is exchanged and an encrypted connection is set up.

When the board loses connection, is powered off or resumes from power-saving mode, it will automatically reconnect without intervention. The stored encryption keys are re-used.

The Windows Bluetooth devices screen should appear as below:



Ultimarc BlueHID

1 item

The Self-Test LED

The self-test LED can be enabled and disabled from within the U-Config utility and should be disabled after installation and testing is complete.

If disabled the LED lights as follows

- Board powered up: LED flashes once.
- Board connected via Bluetooth: LED lights
- Board goes to bind, authorized state: LED goes off

If enabled the LED also blinks at the rate of data polling by the host system This means during normal operation it is seen to be lit constantly, then in idle mode progressively blinks more slowly. This can be used to check that no external device or connection is preventing power saving modes from being invoked.

Note this consumes power so the LED should be disabled in U-Config after installation and testing.

The on-board self-test connected on the board to J1 pin 4. It can be controlled from the host if required.

Section 4: Connecting Devices

Important! Never connect any device to the board which sends more than 5 volts otherwise damage could occur!

Connecting Switches and Pushbuttons

Any pin can be defined as a switch.

You can define a switch as:

- Mouse button (including double-click)
- Gamepad button or POV Hat direction.
- Keyboard key.
- Various multimedia and power control keys

Each switch input has a primary assignment and a secondary assignment (optional).

Both the primary and secondary assignments can be used in key action settings and the secondary assignment is also activated in place of primary when a "BlueHID Shift" key is held down.

You can assign a different action for the event when the switch is closed (ie grounded) and for when the switch is opened.
The two possible keycodes (or buttons) coupled with the way in which you define closed/open events gives a mass of flexibility in the types of control you can connect without using any external circuitry.

The keycode which is sent when you activate the switch is defined using the U-Config utility. You can assign any connection to send any keypress.
Switches can also be assigned as game controller buttons, which are numbered 0-32 (32 is a Windows limitation).
Mouse button selections available are left, right, middle or left double-click.

Example 1: Simple keyboard-type button (Normally Open)

For this you would set the “down” action to be the required keycode and “up” action to be “Clear”.

Example 2: Simple keyboard-type button (Normally Closed)

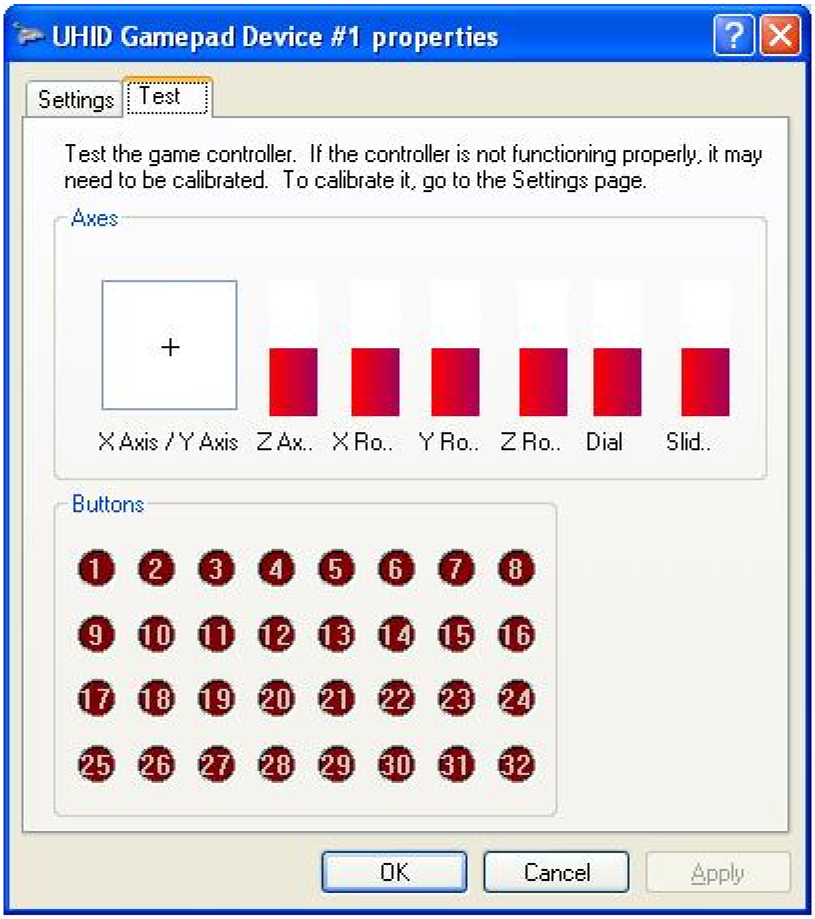
Set “down” action to be “clear” and Up action to be the required keycode.

Example 3: Toggle switch with integral LED

This type of switch sends a 5 volt signal when ON, and is open-circuit when OFF. Let’s say you want to turn on a control when you switch “on” the toggle but you don’t want to generate masses of key-presses which the toggle is still down. You can send one key-press only, by configuring:“down” action to be “clear” and “up” action to be “pulse” and the required keycode. Setting it up this way means the internal LED in the switch will function properly with no external circuitry to control it.

Testing Switches

Switches defined as keyboard keys can be tested using Notepad or any program which displays text.
Gamepad buttons can be tested using Control Panel, Game Controllers. The display is shown below:

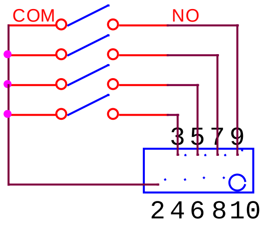


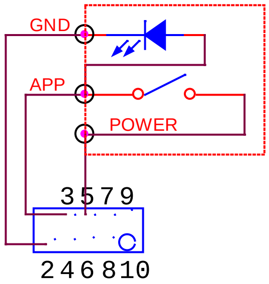
UHID Gamepad Device #1 properties

Settings | Test

Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

Examples of Switch Connections

EXAMPLE: FOUR PUSHBUTTON SWITCHES			
		PIN	CONFIG
		2	GND (FIXED)
		3	SWITCH (KEY OR BUTTON)
		4	
		5	SWITCH (KEY OR BUTTON)
		6	
		7	SWITCH (KEY OR BUTTON)
		8	
		9	SWITCH (KEY OR BUTTON)
		10	

EXAMPLE: TOGGLE SW WITH INBUILT LED			
		PIN	CONFIG
		2	GND (FIXED)
		3	SWITCH (KEY OR BUTTON)
		4	
		5	5 VOLTS
		6	
		7	
		8	
		9	
		10	
NOTE: For this type of switch, the switch must be configured as: DOWN = CLEAR UP = <REQUIRED KEY>			

Connecting Analog Devices

Important! Never connect any device to the board which sends more than 5 volts otherwise damage could occur!

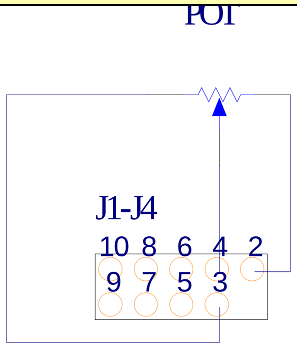
A pin assigned to Analog senses a voltage on the pin and controls a gamepad X, Y, or other axis on the PC.
Analog resolution is 12 bit.
The voltage can vary between 0 and battery+ volts.
Usually this voltage will come from a potentiometer (pot) which is part of a steering wheel, handwheel, throttle pedal, or other type of analog control.
You can assign any pin to one of 6 possible analog axes.

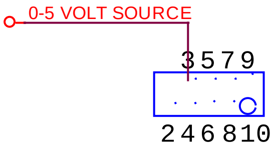
The first two analog axes (X and Y) can be configured in U-Config to have an adjustable "dead zone" in the centre of movement and also for auto-centering at initialization.

To test in Windows, go into "Control Panel", "Game Controllers". Then select the board number (if you have more than one board).You will see the X, Y crosshair (axes 1 and 2 on the board) and also the other axes as red bars.

Using high-value pots (100K ohms or greater) will consume less power and extend battery life.

To prevent battery drain the positive side of the pot should be connected to a pin defined as 5V. These pins "float" when the board is in power-saving modes thereby un-powering the pot. The negative side of the pot can be connected to any ground fixed pin.

EXAMPLE: POTENTIOMETER AS AN ANALOG CONTROL			
		PIN	CONFIG
		2	GND (Fixed)
		3	5V
		4	ANALOG INPUT
		5	
		6	
		7	
		8	
		9	
		10	

EXAMPLE: VOLTAGE SOURCE AS AN ANALOG CONTROL		
	PIN	CONFIG
	2	GND (FIXED)
	3	ANALOG INPUT
	4	
	5	
	6	
	7	
	8	
	9	
	10	

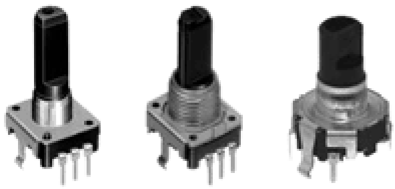
Connecting Optical or Mechanical Encoders (Spinners, Trackballs etc)

High-resolution Quadrature devices usually consist of a pair of optical sensors and an encoder wheel. Examples of such devices are:

- Spinners such as the Ultimarc SpinTrak
- Trackballs such as the Ultimarc U-Trak
- Incremental Optical Rotary Encoders
- Incremental Mechanical Encoder eg CTS 288 Series

- 12-way rotary switches such as the Happ Controls Rotary Joystick. See App Note AN1001

- Linear Motion Sensors



Mechanical Encoders

Ultimarc U-Trak

Ultimarc SpinTrak

Mouse Mode

Quadrature devices can be used to control the mouse pointer in an X, Y or Z direction. 3 high-resolution devices are supported which must be connected to opposite pins on J1 or J2.

Button Pulse Mode

Note: This mode is suitable only for low-resolution mechanical rotary encoders. 8 encoders are supported which must be connected to opposite pins on J1 or J2.

In this mode, the quadrature device is assigned to a pair of Game Controller buttons. The pairs are buttons 1,2, buttons 3,4, buttons 5,6 etc. When the encoder is turned in one direction, the first button will be repeatedly "pressed". When turned in the other direction, the other button of the pair will be repeatedly "pressed".

Connections

Quadrature devices use two input pins per device, and optical devices will require power. Mechanical encoders require GND only, and this is the center pin on 3-pin devices.

To prevent battery drain the positive side of the encoder power should be connected to a pin defined as 5V. These pins “float” when the board is in power-saving modes thereby un-powering the encoder. The ground side of the encoder can be connected to any GND pin.

Windows allows one instance of each axis when defined as Mouse (ie X, Y, Z) but if you are using multiple boards, Windows can recognise each board as a different mouse, using DirectInput.

Devices should be capable of operating with the battery voltage supply (not required for mechanical encoders).

Note that only certain pins can be configured for quadrature devices. These can be seen in U-Config by clicking on the "Quadrature" line in the PCB image. Appropriate pins will be highlighted. Two pins are used per device. An X-Y device such as a trackball counts as two devices and therefore uses 4 pins plus power.

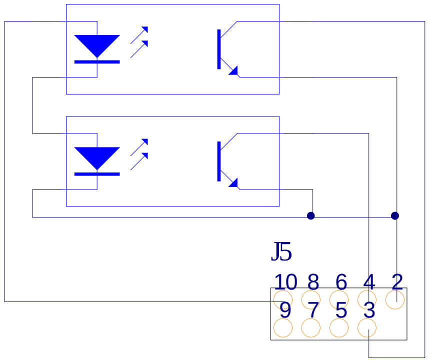
Mechanical Encoder Types

The type of mechanical encoder suitable for this application are the types with a defined logic output on each detent position. These include the CTS 288 series. Some encoders such as the Alps EC12 have undefined outputs at the detent position and are not suitable where a single “click” per detent is required.

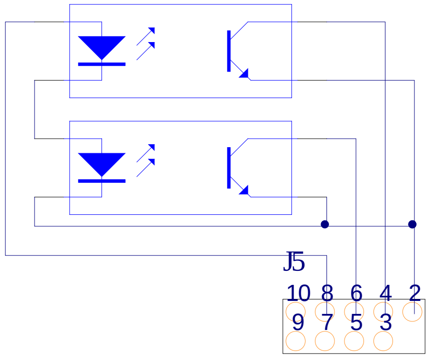
Ultimarc U-Trak Trackball (Special Case)

This device can be directly plugged into the board, using the J5 connector and is automatically powered in the correct way, using power which is switched off when in power-saving modes.

Examples of Quadrature Device connections

EXAMPLE: OPTICAL WHEEL (SPINNER) using J1 or J2 (Inputs on any pair of pins)			
		PIN	CONFIG
		2	GND (Fixed)
		3	X1
		4	X2
		5	
		6	
		7	
		8	
		9	
		10	5V

The example below uses the "Switched Ground" output pin to supply the encoder with sleep-controlled power instead of using a 5V pin.

EXAMPLE: OPTICAL WHEEL (SPINNER) using J5 (Input pins 4-6 or 3-5)			
		PIN	CONFIG
		2	Switched GND (Fixed)
		3	
		4	X1
		5	
		6	X2
		7	
		8	5V (Fixed)
		9	
		10	

Connecting LEDs or other output devices

LEDs or outputs can be controlled either using our SDK from the host or locally from control input. Note that an illuminated LED will seriously reduce battery life.

The BlueHID will drive standard LEDs only, max current 20 mA

LEDs will need a series resistor, or use 5 volt LEDs with an in-built resistor.

LEDs are connected between the pin defined as output, and a ground pin.
The pre-made modular harness can be used to connect LEDs.

When in power-saving modes, all LEDs will be turned off and outputs will “float”.

Other devices can be connected but the following applies:
Max current draw: 20 mA
No external voltage must be applied to the controlled device.

BlueHID Controlled LEDs

BlueHID Controlled LEDs are LEDs which are turned on or off by switches which are connected to other pins on the board.
Up to 16 connections can be assigned as BlueHID-controlled LEDs. When you assign these LEDs, you link it to any other connection. The other connection must be defined as a switch. This other connection will control the state of the LED.
Its important to realize that the LED state is not actually controlled by the other switch connection pin directly, but by the boards stored internal state of the control connected to that pin. This means the control of the LED is affected by how the switch is defined as well as its actual open/closed state.

Example1

Assume you have a pushbutton which has an internal LED. It is a momentary button so cannot be used to directly switch the LED since it would only light while the button is pressed.
You have this button configured as a toggle (push on, push off).
You can assign the LED to this button, and it will light when the button is pressed and go dark when it is pushed again, in sync with the buttons action in the application.

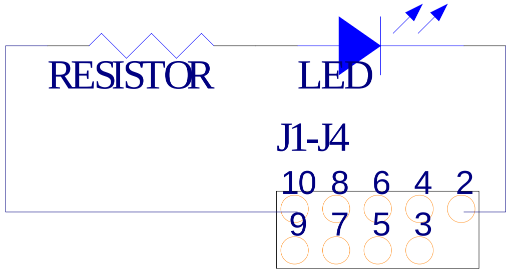
Example 2

Assume you have a toggle switch which grounds one connection when closed. This could not be used on its own to also control a LED.You could link a BlueHID controlled LED to this connection and it would light while the switch s closed.

PC Controlled LEDs

Up to 16 connections from all pins can be defined as PC-Controlled LEDs. You can also use these for other output functions.
These LEDs are controlled by a software application on the PC. There are sample APIs for developing your own applications to add LED or output control.

You can also assign LEDs to behave as keyboard CAPS, NUM and SCROLL lock LEDs. They will be turned off or on in the same way as your regular keyboard LEDs.

EXAMPLE: LED (STANDARD TYPE) Any pin.		
	PIN	CONFIG
	2	GND (FIXED)
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	LED OUTPUT

Connecting to Power and Ground (Configurable pins)

You can assign any connection as power output or ground.

Power and ground pins can only source/sink 20mA max current per pin.If you need more current than 20mA then you can use the fixed power pins on connector J5 (see below).

Power pins are switched off and go to a floating state when in power-saving modes, to conserve battery power.

IMPORTANT: In addition to turning off during power-saving, pins defined as 5V pulse at varying rates during normal board operation. This is to minimize power consumption of attached devices and is part of the advanced power-saving functionality of the board.

Connecting to high-current Power and Ground (Fixed pins)

J1 J2 J3 J4 pin 1
These pins are directly connected to battery negative

J5 pins 7, 8
These pins are both connected to battery positive via the reverse-polarity protection circuit. These are not turned off when in power-saving modes.

J5 pin 1
This is a special pin which has high current-sink capability of 100mA and is turned off when in power-saving modes.

External devices which require more than 20 mA should be powered from J5 pins 7 or 8 (positive) and J5 pin 1 (negative). The device(s) will be powered down when in power-saving modes.

Section 3: Configuration

Understanding The BlueHID Shift Feature

The internal shift feature allows you to define a connection (or connections) which when activated, cause all other switch connections to send an alternate command. If you only want each pin to send one key or button, you can skip this section.

This is rather like a keyboard shift key which, when held, causes the keyboard keys to send different characters. When the internal shift connection is activated and held, the Secondary function of all other keys is sent instead of the Primary.

EXAMPLE 1:
In a gaming controller you might wish to assign an admin function such as COIN INSERT and you don't wish to have a dedicated button on the panel for this. You could make PLAYER1 START an internal shift connection. Then assign a secondary COIN INSERT function to the switch which is normally "PLAYER 1 FIRE". So to insert a coin, hold PLAYER1 START and press PLAYER1 FIRE.

EXAMPLE 2:
You have a control panel which is used for two completely different functions and you wish these functions to be selectable using a toggle switch. You could assign a spare connection as an internal shift connection, and then connect a toggle switch to it. Then assign all Primary functions to your first control panel type and all Secondary functions to the second type. Switching the toggle would flip the panel from one mode to the other. (Be aware you can also achieve this by re-programming the board on-the-fly though!).

Defining Internal Shift Connections

Normally you would perform this on only one connection, which will act as the shift key.

Most users would only require to define this function once, so to avoid clutter on the display, it is hidden unless the connection is completely un-assigned. This means with the default configuration, which has all connections assigned, you won't see this option.

To assign a connection as an internal shift:

Click on the connection and select “not used”. A check box will appear with the option “This is an Internal Shift Switch”.

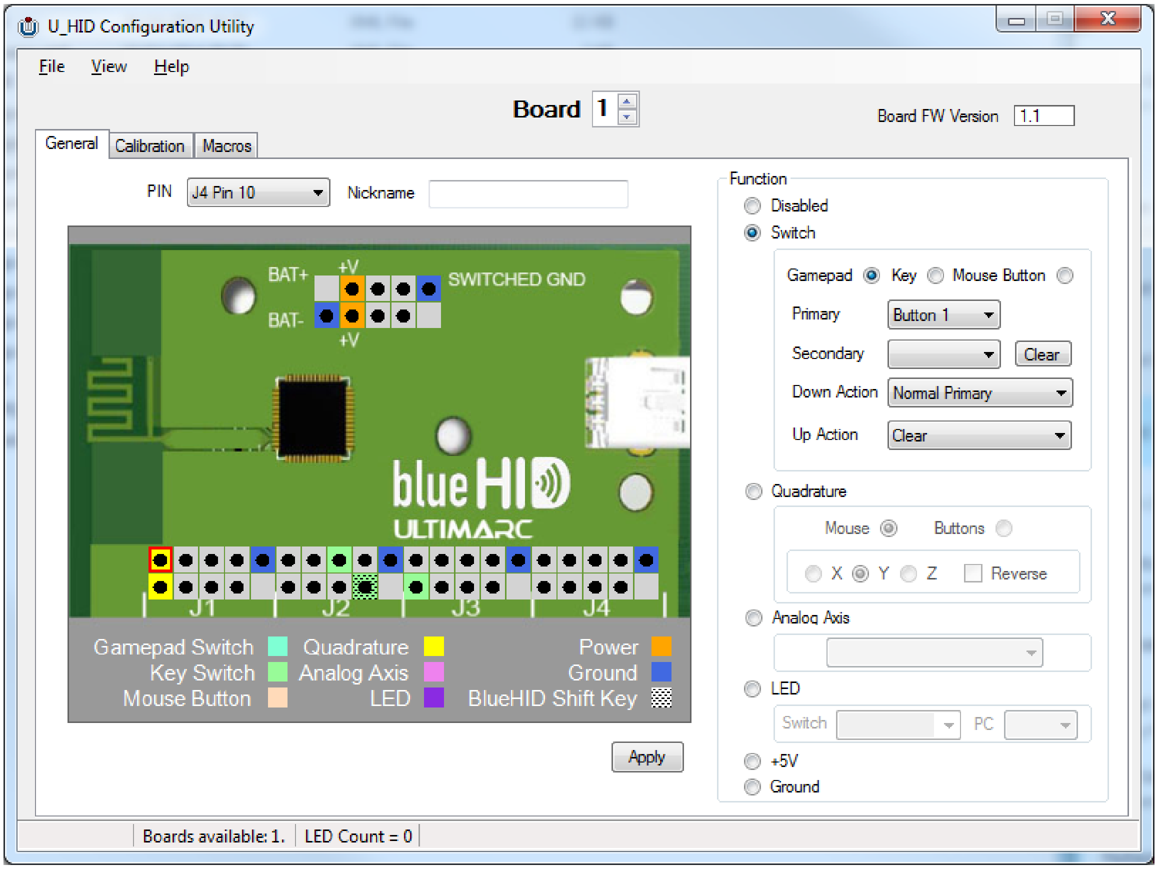
Enable the check box. This will be highlighted in red and will remain visible while this connection is selected.

Checking this box limits what you can do with the other selections for this switch. You are limited to assigning a “Pulse action on Open” or no action at all. Any other type of assignment is invalid for an internal shift connection.

You cannot assign this connection as an LED, Analog or Quadrature input.The ability to assign a “pulse on open” action means you can also have this connection perform its own function as well as being a shift key.

If the key is pressed and released, it will send its own function as a pulse, provided no other connection has been activated in the meantime. This is how Example 1 works, above. The Player 1 Start function of this key still works even though it’s a shift key as well. You can assign any number of the connections to be internal shift, but they all will enable the same shift functionality in addition to their own individual assigned code (if any).

Using the U-Config Configuration Utility



U_HID Configuration Utility

Board 1

Board FW Version 1.1

PIN J4 Pin 10

Nickname

Function

○ Disabled

● Switch

Gamepad ● Key ○ Mouse Button ○

Primary Button 1

Secondary

Down Action Normal Primary

Up Action Clear

○ Quadrature

Mouse ◉ Buttons ○

○ X ◉ Y ○ Z □ Reverse

○ Analog Axis

○ LED

Switch PC

○ +5V

○ Ground

Boards available: 1. LED Count = 0

This utility is used for the following: .

- Configuring each pin of the board as the correct device type, ie keyboard, gamepad, analog, quadrature mouse, power.
- Configuring each pin with the correct keycode, button number, axis or other parameter
- Assigning macros
- Configuring analog axis scale factor and offset
- Configuring power-saving features.
- Loading and saving configurations
- Initiating a firmware upgrade
- Downloading a configuration "on the fly" using a command-line task

The graphical picture of the board shows the way in which the pins are divided into sections, each being one 9-pin connector (plus key). Pin 1 is the key and pin2 is ground. The utility interacts with the board in real time, every time a change is made. The configuration is stored in flash ROM on the board so it remains after power off/on. When the utility is run, it reads the configuration from the connected board and displays it.
You can also store the entire configuration as a file for later downloading on-the-fly for multiple configurations.

Function of each GUI feature:

GENERAL tab (main window)

PIN drop down
This displays the selected pin you are configuring. You can select pins from here or from the main graphic.

NICKNAME
You can type text in here which will remind you of which device this pin is connected to. The test is saved with the configuration (on the PC only) and is not sent to the board.

FUNCTION area
Here you configure the selected pin as the required device type and assign values to it.You can select one of the radio buttons to define the pin as the following:

Switch	Includes keyboard key, gamepad button and mouse buttons
Quadrature Mouse	Includes trackballs, optical steering wheels etc
Analog	Gamepad type control
LED output	Includes BlueHID and PC Controlled
+5 Volts out	Max 20 mA supply. Pin disconnected when in power-saving modes
Ground.	Max 20 mA sink. Pin disconnected when in power-saving modes. Most connected devices will use the fixed GND pins though.

APPLY button

This button applies the changes you have made to the pin. Changes are immediately sent to the board.

GRAPHIC AREA

You can click on any pin on the graphic to start configuration of this pin. Additionally, if you click on the pin types listed in the center of the PCB, this will show which pins are allowed to be configured as this type of control.You will notice that certain pins on P5 are fixed. These are high-current power/ground pins and are hard-wired and cannot be changed.
Each 10-pin connector area has one hard-wired ground pin.

CALIBRATION tab

This tab allows you to set scale factor and offset for each analog input. The default values give a full-scale travel with the cross-hair centered at mid-position of the pot. It may be necessary to experiment with these settings and you can observe the result by opening Control Panel, Game Controllers in Windows, and clicking on the came controller device to display the calibration screen.

Detail Configuration Example (Switch)

The screenshot above shows this configuration.
We will look at how a switch is connected and configured.
Firstly, the switch is connected with one terminal routing to the appropriate pin and the other to a ground pin.

- Select the pin by clicking on it in the graphic area.
- Select the "Switch" radio button in the Function area.

You now need to select what type of control this will be, whether a mouse button, keyboard key or gamepad button.

- Select keyboard key.

In the Primary drop-down, all possible keyboard keys will be available, plus macros. Macros are defined separately, see later.

- Select which keyboard key you need.

Note that the BlueHID sends key scan codes, just like a keyboard. It does not send characters so has no concept of upper/lower case. An upper-case character is two keys (ie shift and the character key). There is no need to assign a secondary code unless you wish to use the shift feature or other special functions, see later.

Then, you need to tell the board how and when to send the keycode you have chosen. This is done with the Down and Up Action drop downs. These can be selected as follows:

Clear

This removes the keycode associated with this pin from the buffer, so it is no longer sent

Normal Primary

This sends (and holds) the primary keycode you selected above, to the PC. If the BlueHID Shift button is pressed, it will send the secondary code instead.

Normal Secondary

This sends (and holds) the secondary keycode but you have not assigned one at this point so no action.

Pulse Primary

This sends a short pulse, simulating the key being pressed then released, even though it might still be physically closed (eg a toggle switch left closed)

Pulse Secondary

This is the same as above but pulses the secondary code.

Toggle

This is not explicitly an option, but is the result of setting both the Down and Up actions to Normal Primary. This will cause the selected keycode to alternately be sent or not sent on alternate button presses.

Flip-Flop

This is not explicitly an option but is the result of setting Down action to Secondary and Up action to Primary. This will cause the primary or secondary key to be sent on alternate button presses.

See-Saw

This is not explicitly an option but is the result of setting Down action to Primary and Up action to Secondary. This will cause the primary key to be sent on press and secondary on release. Normal or Pulse mode can be used.

The most common usage of keyboard keys is to set Down Action to Normal Primary and Up Action to Clear.

Important Note about keyboard auto-repeat:
On HID keyboards (and therefore the BlueHID) the auto-repeat of the keyboard is done in Windows. So if you assign any keycode to be pressed permanently (which can occur in many configurations) you will get an endless string of characters in your application. For this reason, many of the advanced combinations should only be used with gamepad buttons, which do not have a problem being held in the pressed position.
You should also avoid defining any keyboard keys from being "pressed" by default as this will cause boot failures (stuck key) and may cause unexpected problems in the Windows GUI.

Other switch types:

Mouse Button
You can select left, middle, right and left double-click from the drop-down.

Gamepad Button
These are numbered 1-32 as shown in "Control Panel, Game Controllers".

Quadrature Mouse Selection

This selection always uses two pins. When you assign the pin as this device type, the opposite pin will also automatically be assigned.
You can select X, Y or Z (wheel) axis.
Trackballs use two axes (4 pins).

Quadrature Button Selection

This selection uses two pins. When you assign the pin as this device type, the

opposite pin will also automatically be assigned.
You can select one of 8 pairs of game controller buttons which will be pressed repeatedly when the encoder is turned. Note you can configure the "press" time on the Calibration tab.

Analog Axis

These are numbered 1-6. You can view analog axes using Control Panel, Game Controllers.
The first two axes are displayed as X and Y.

Conventional analog axes

The analog behavior can be fine tuned by adjusting the offset and scale factor in the calibration window (tab on the top of the display). To observe changes, open the Game Controller window in Windows. In Win7 and later this is accessed by “Devices and Printers” and right-click on the BlueHID device.

Analog X Y with deadzone and auto-calibration

Normally, the host will see a linear change in the analog value for each axis. But the X and Y axes can be configured in the drop down to emulate a game controller which has auto-centering at power-up and a dead-zone of no movement in the center of travel.
In this mode, the amount of dead-zone is adjustable by changing the “offset” value in the calibration window (tab). The scale factor may need to be adjusted for full travel.

LED Output

There are two types of LED. Pins defined as LED are switched off when the board is in power-saving modes to conserve battery power.

BlueHID controlled (switch)

This type of LED is switched on when the switch allocated to it is sending its keycode or button (which might not necessarily mean the switch is physically closed, depending on how the switch pin is configured).
In the drop-down, you can select which switch (referenced by its pin) controls this LED.

PC Controlled LED

This type of LED is switched on/off by a PC application. These are numbered 1-16.

Macros

Macros are sequences of key-presses. Each key is logically pressed in turn, and held, then after the last key in the macro is sent, all are released. This means for example, the sequence "Ctrl-Alt-Del" can be programmed.

Up to 30 macros can be programmed and there is a limit of 84 macro keys (total of all keys in all macros).

Macros are first set-up using the Macro tab on the main U-Config screen. Each macro is assigned a number. Then the macro number is used in the main configuration.

For example to configure "Ctrl-Alt-Del" as a macro:

1. Select the Macros tab in the main window
2. Click "New"
3. From the drop-down, select L-CTRL
4. Click "Add Entry"
5. Repeat steps 3 and 4 for "L-Alt" and "Del"
6. Click "Add macro"

The macro will now be shown in the list and a number is assigned to it. You can now use this macro number in the main configuration screen in the same way as any other keycode etc.

Other Functions

FILE menu

Open, Save

These selections allow loading/saving of .HID configuration files, so you can have multiple saved configurations.

Import/Export

You can import and export configurations to a CSV or XML file so you can edit them outside of the application or generate using a different application.

Firmware Upgrade

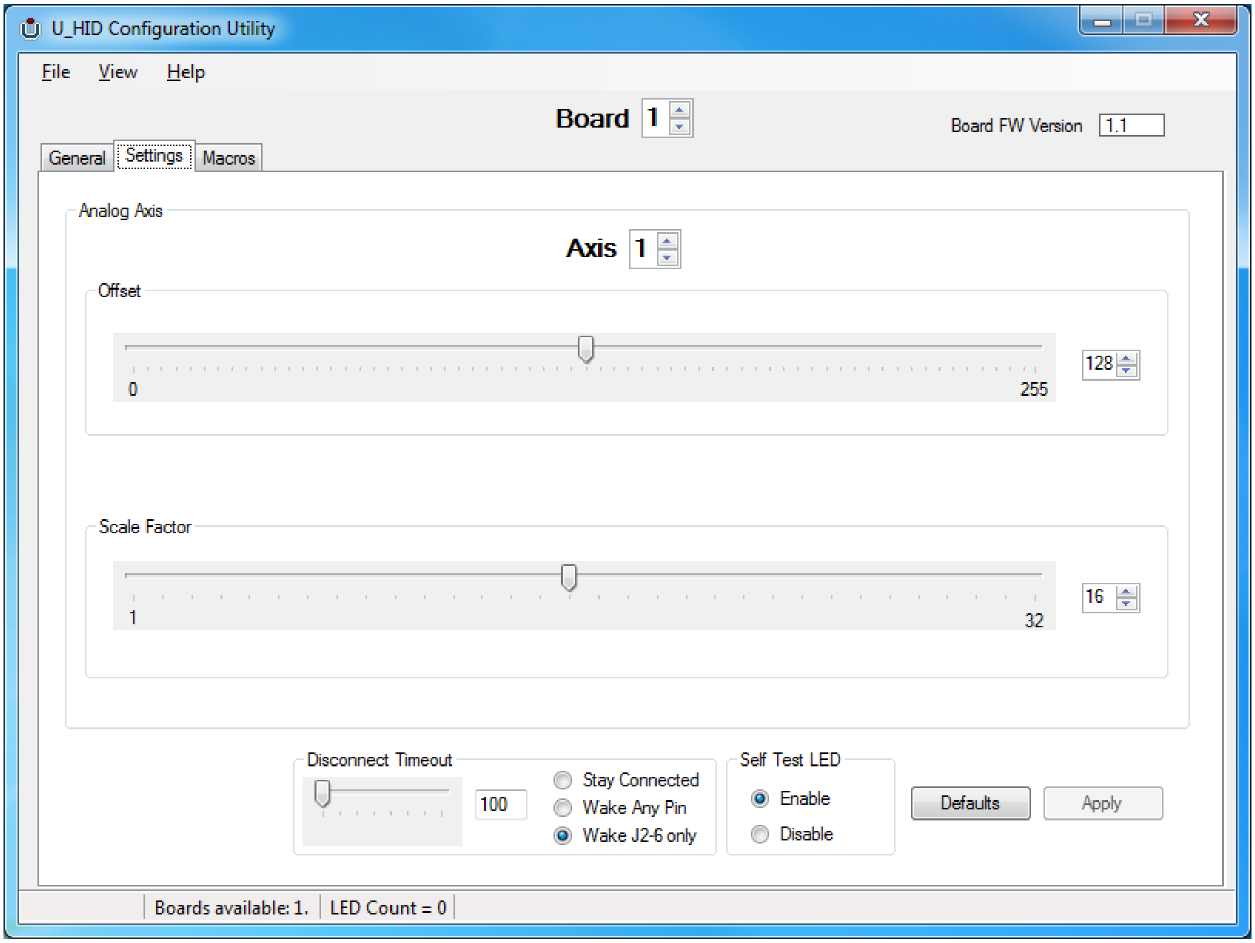
This initiates a firmware upgrade. The board assumes a different Bluetooth ID and a driver will be installed.

VIEW menu

There is only one option, "View BlueHID Shift Option". This is normally hidden in the main area to avoid clutter.
This is only used when assigning a pin to be a BlueHID shift key.
The shift option allocates a key (or keys) which, when held, cause other pressed keys to send their secondary instead of primary keycodes. This over-rides the Primary/Secondary Action defined for keys.

Power Saving Mode Configuration

These settings are located on the “Configuration” tab



Board 1

Board FW Version 1.1

Axis 1

Offset 128

Scale Factor 16

Disconnect Timeout 100

○ Stay Connected
○ Wake Any Pin
● Wake J2-6 only

Self Test LED
● Enable
○ Disable

Boards available: 1. | LED Count = 0

The slider defines how many seconds of no-activity elapse before the board disconnects the Bluetooth link. Or you can opt to have the board never disconnect but this will consume more battery power.

***Wake Any Pin* button**

After disconnection, the board will wake and re-connect if activity is detected on any pin which is defined as a switch.

Wake J2-6 Only **button**
When this is selected, the board will only wake when activity is detected on pin J2-6. In this mode (Stop Mode) the board is in maximum power-saving state.

Self-Test LED
The self-test on-board LED can be disabled. This is recommended after testing is complete, to conserve power.

Running from a command-line

The U-Config utility can be run from a command-line (ie Batch file) by using the command “U-Config <name of HID file>”. The board will be programmed with the contents of the HID file.

Section 4: Output Software API

Introduction

The Ultimarc PacDrive & U-HID SDK is a collection of source code examples for controlling the PacDrive & U-HID LED controller hardware by Ultimarc.
It can be downloaded from:
http://www.ultimarc.com/programming.html

It contains source code projects for the following languages:

- C#
- C++
- Delphi
- VB6
- VB.NET

API Functions

int PacInitialize()

- Initialize all devices
- Returns the number of boards connected to the PC

void PacShutdown()

- Shutdown all devices
- No return value

bool PacSetLEDStates(int id, USHORT data)

- Sends data to the board specified by id
- Returns true for success and false for failure

Section 5: Technical Data (Dimensions in mm)

W	33
L	60
H	10
Mounting Hole Dia	3
Operating Temp deg C	-10 - +85
Supply Voltage Range	1.8 - 5.5 volts
Max voltage allowed on inputs	Supply plus 0.5 volts
Normal supply voltage	3.3 volts
Active Supply Current (approx)	4 mA
Sleep Supply Current all pin wake	300 nanoamps
Sleep Supply Current wake on J2-6 only	60 nanoamps
Pull-Up Resistors (except when configured as analog input)	5.6 Kohms
Number of pins available (total)	45
Number of inputs available	36
Number of host-controlled outputs available	16
Max output voltage when OFF	0.75 V
Max output current per pin	18 mA
Max output current per board	200mA
Number of analog axes	6
Analog Resolution	12 Bits
Input Impedance	10 Mohms
Input voltage swing	GND to Supply
Number of quadrature pins available	10 pairs
Number of axes	3 (X, Y, Z)
Input DC requirements	Source drives low. Pulled up via 5.6 Kohms
Encoder type	2 bit gray code (quadrature)
Host interface resolution	16 bits
De-Bounce lock-out of same pin (switch inputs)	20 ms
Max latency	10 ms

Max number of macros	30
Max number of total keys in macros	84
Bluetooth Version	4.1
Security Specification	Mode 1 Unauthorised Pairing with Encryption.

Connectors

The open pin headers can be used with a variety of 0.1 in (2.54mm) connectors. These have the advantage of being end and side-stackable. Example housings are Molex p/n 22-55-2401 (40 way) or 22-55-2101 (10 way)
Crimp terminals for these are 16-02-0102.
A partial-length connector can be used in any location, for example 10-way, 12-way etc will fit.
Note a selection of ready-made harnesses are available from Ultimarc.com

Section 6: Upgrading Firmware

Note the “Over The Air” firmware update procedure has been tested using the CSR 4.0 dongle with CSR-supplied host software, on Windows 7 and also without any host software on Windows 8 and 10.

The “Upgrade Firmware” button under “file” causes the board to change to a different Bluetooth device and reset.

The firmware upgrade procedure is as follows:

1. Ensure the board is connected in normal mode and not in power-saving mode (make it active by using a control if necessary)
2. Run the U-Config program
3. Select “File, Upgrade Firmware”
4. Accept the OK prompt.
5. Open the “Bluetooth devices” dialog in Windows
6. Search for new devices. An “Ultimarc Loader” device should be found.
7. Click on the new device to bind to it.
8. If not already running, go to the U-Config install folder and run the program called “Blueupload.exe”.
9. Select the new firmware file from this program and the upgrade should start.

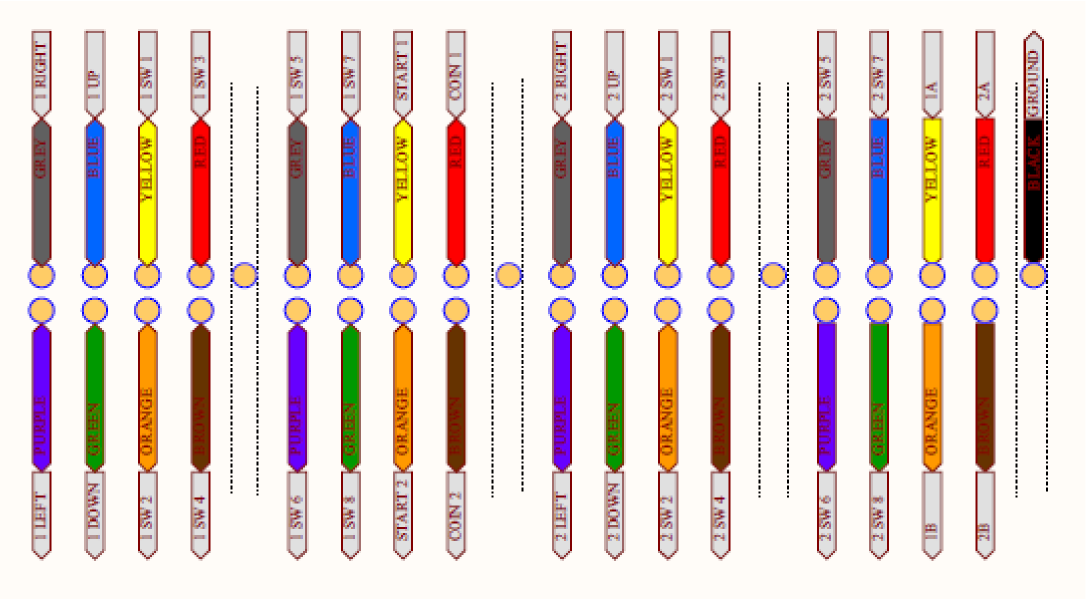
Section 7: Host Interface Data Specification

HID Keyboard	Input	All available keys
HID Keyboard LEDs	Output	Caps, Num, Scroll lock LEDs
HID Mouse X	Input	16 Bit X displacement
HID Mouse X	Input	16 Bit Y displacement
HID Mouse X	Input	16 Bit Z displacement
Mouse Buttons	Input	5 buttons
HID Joystick	Input X	12 Bit value
HID Joystick	Input Y	12 Bit value
HID Joystick	Input Z	12 Bit value
HID Joystick	Input RZ	12 Bit value
HID Simulation Control	Input Brake	12 Bit value
HID Simulation Control	Input Accelerator	12 Bit value
HAT Switch	Input	4 combinational directions
Game Controller Buttons	Input	32 Buttons
Consumer Control	Power	
	Sleep	
	Volume Increment	
	Volume Decrement	
	Back	
	Home	
	Play	
	Previous	
	Stop	
	Eject	
LED Output	Output	16 Outputs

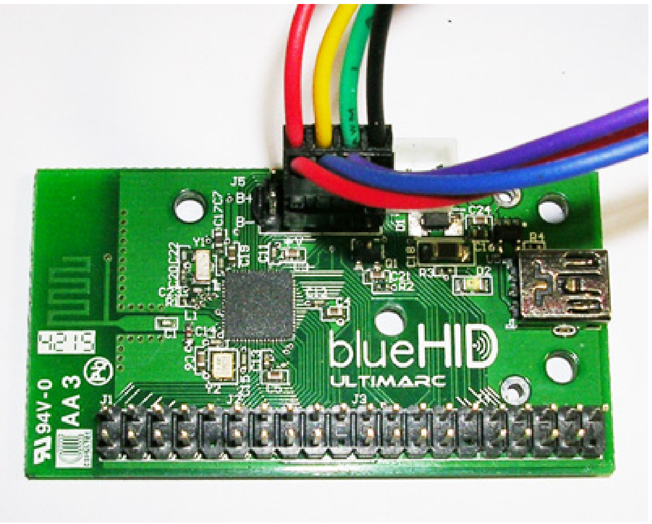
Section 8: I-PAC Pre-Configured Mode

The I-PAC configuration is a pre-configured version optimized for MAME use, with a 32-way switch harness (U-HID Harness Type 3)

The connector pinout and harness wire colors are shown below:



The Ultimarc U-Trak trackball can be connected to the top header as shown below:



NOTE: The current version of the U-Trak trackball requires the board to be powered from a supply voltage of 4.5 volts, ie 3 X 1.5 volt batteries or 3 x rechargeable batteries. For this reason the Battery Holder version of the board is not suitable when a trackball is being used. Lithium-polymer battery will not supply a high enough voltage for use with the trackball.

We will have trackball versions which will operate from LiPo battery during 2016.

The Ultimarc Spintrak spinner will operate from a LiPo battery. This is electrically compatible with this board although some connector re-pinning will be necessary as there is not a dedicated connector provided for this device.

Provided the board is configured for “Wake on P2-5 only” with a relatively short sleep timeout, there is no need for a power switch in this configuration. The trackball is powered off when in power-saving mode and therefore will not wake the controller.

Section 9: Wiring Harness Options

There are 3 types of pre-made wiring harness available in our online store:

U-HID Harness Type 3

- 32 connections to female blade connectors.
- Available with 4.8mm, 6.3mm or a combination of 2.8 and 4.8mm connectors.
- Length 350mm

U-HID Modular harness

- 9 wires. Double ended harness which can be cut into two. Length 1000mm
- Connectors fit onto one section of pins (8 connections plus ground).

Extension Harness Packs

600mm long wires which replace standard wires in the Type 3 harness for longer wire requirements.

Daisy-Chain Harnesses

For ground daisy-chain to multiple switches. 40 connections pitched at 100mm distance.