1400 Series Audio-Nav Keypad
Technical Manual

RNIB Tried and Tested

Product Features

AudioNav is an ADA compliant assistive USB device offering menu navigation by means of audible content description.

Users with impaired vision, reading difficulties or impaired fine motor skills can navigate through menus or directories that would typically be presented on a visual display or touch screen. Screen content is represented and summarised by recorded or synthesized language via a headset or handset.

This provides a set of menu selection keys which are differentiated in a way that makes the product easier to use by people with visual impairment. In addition, a standard 3.5mm headphone socket is provided. This allows customers to plug their headset into the module and receive audio instruction to help them navigate the use of the equipment.

The externally mounted version of the AudioNav provides options for manufacturers and operators to permanently affix an AudioNav device to the outer casing of a host terminal or to adjacent surfaces such as walls or service counters. This is especially useful when existing self-service installations must be upgraded to meet current accessibility mandates.

An optional ‘Quick Release Cradle’ allows the AudioNav to be detached from the host system for use as a hand-held device. In this hand-held configuration AudioNav can, if required, be passed directly to any user with limited reach or impaired dexterity.

Used in combination with SpacePole™ products this externally mounted version of the AudioNav can be conveniently positioned and adjusted to ensure maximum accessibility.

By use of the utility software, default illumination status and ‘wake-up’ behaviour can be selected. The USB codes can also be changed. Connection to the host is via a single USB cable.

Keypad

- Keypad is available in underpanel or externally mounted versions, with the following keys :
- A 4 way directional key providing UP, DOWN, LEFT and RIGHT navigation.
- A central ENTER key
- An illuminated audio volume key
- Illuminated 3.5mm audio jack socket (illumination under software control)
- Orientation switch in underpanel version to allow portrait or landscape mode.
- Mini-USB socket for connection to host (external version has fitted cable)

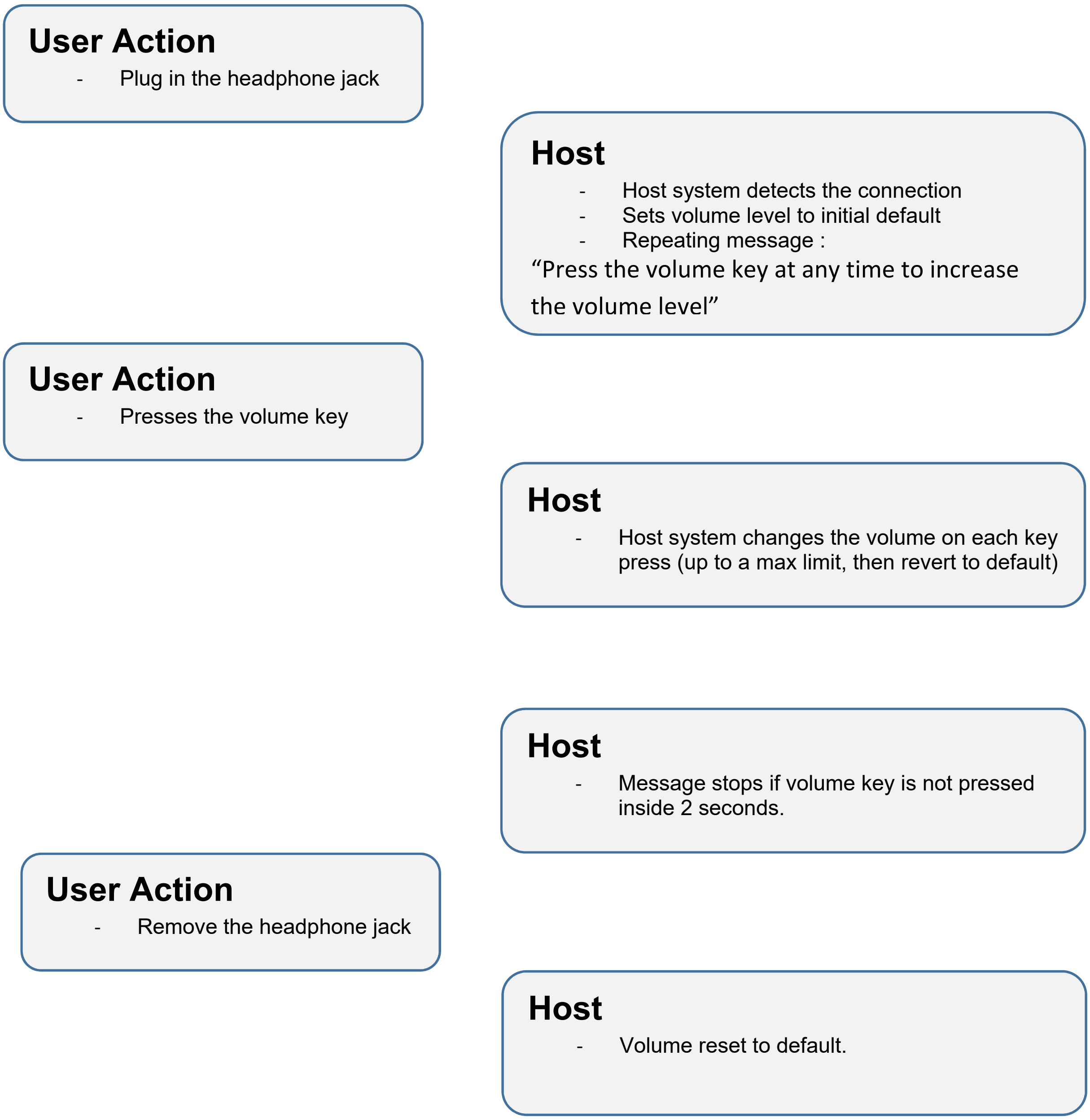
USB Interface

- HID keyboard
- Supports standard modifiers, i.e. Ctrl, Shift, Alt
- HID consumer controlled device
- Advanced audio device
- No special drivers required
- Audio Jack Insert / Removal sends USB code to host

Support

- Windows compatible utility for changing the USB Code Tables
- API for custom integration
- Remote Firmware update support

Typical method for audio module volume control using the API



User Action

- Plug in the headphone jack

Host

- Host system detects the connection
- Sets volume level to initial default
- Repeating message :

"Press the volume key at any time to increase the volume level"

User Action

- Presses the volume key

Host

- Host system changes the volume on each key press (up to a max limit, then revert to default)

Host

- Message stops if volume key is not pressed inside 2 seconds.

User Action

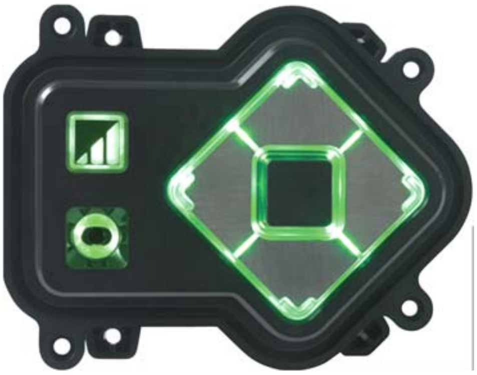
- Remove the headphone jack

Host

- Volume reset to default.

Underpanel Version

Part Number 1406-33001 6 KEY DEVICE + USB AUDIO

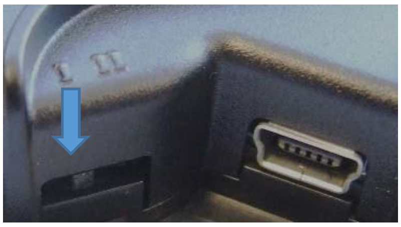


The Audio-Nav is for underpanel use in either portrait or landscape orientation. There are 2 sets of fixing lugs :

- for weld studs on steel panel (1.2mm – 4mm thick), and
- for threaded inserts on plastic panel (3mm thick).

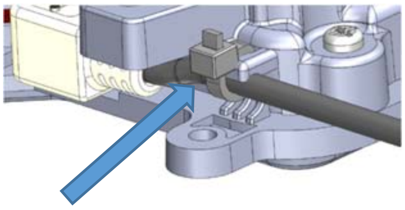
An orientation switch is provided so that the keypad can be fitted in portrait or landscape orientation.
This sends a USB code to the host : factory default is landscape

(Landscape = switch position I shown in picture)



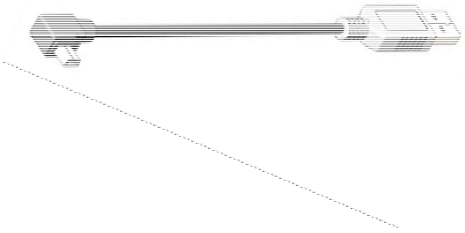
The keypad is designed to be installed underpanel onto M3 weld studs.
Download CAD File for panel cutout drawing.

It is recommended to use a cable tie for strain relief on the USB cable.
(Use 2.5mm nylon cable tie, RS 233-402 or equivalent)



Accessories / Cables

4500-01 USB CABLE MINI-B TO TYPE A, 0.9m



Externally Mounted Version

Part Number	1406-33002	6 KEY DEVICE + USB AUDIO (includes 2m Cable)
	1406-QR000	Quick Release Bracket Kit (includes Qty 4 T20 M4 x 10mm screws)

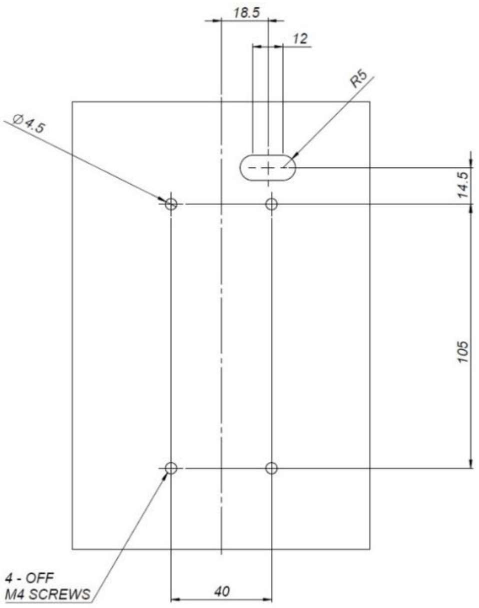
The externally mounted Audio-Nav is for use either fixed directly to a panel, or on a stand.

For direct panel fixing use M4 screws through the panel into the brass inserts on the rear of the Audio-Nav

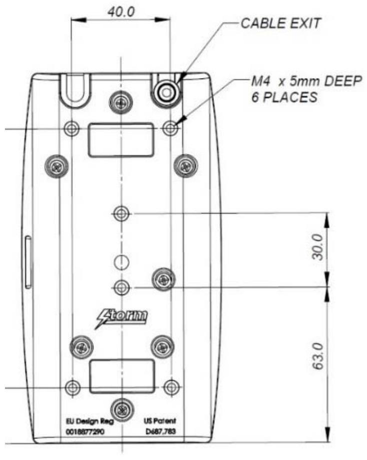
If used with a Spacepole stand then use the Quick Release Bracket kit

Compatible with Spacepole Stack STP101-02

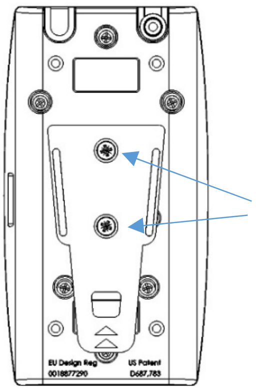
Panel Detail



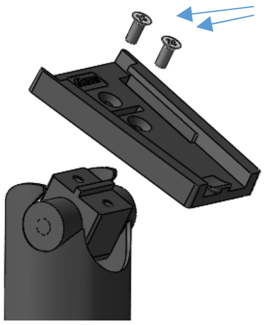
Rear View



Rear View with QR Kit



Fit clip to Audio-Nav with 2 screws



Fit bracket to stand With 2 screws

Specifications

	Underpanel	Externally Mounted
Rating	5V ±0.25V (USB 2.0)	5V ±0.25V (USB 2.0)
Connection	mini USB B socket	USB A Male 2.0
Compatibility	Windows 10 / Win 8 / Win 7	Windows 10 / Win 8 / Win 7
Audio	3.5mm jack socket illuminated	3.5mm jack socket illuminated
Audio Output level	30mW per channel max into a 32ohm load	30mW per channel max into a 32ohm load
Ground	M3 thread grounding point	
Dimensions	Overall 105 mm x 85 mm	Overall 150mm x 82mm x 34mm
Underpanel Depth	25mm	
Cable	Not Included	2 M (includes coiled section)

Mechanical

Key Operational Life	1 million

Performance/Regulatory

Operational Temp	-20°C to +70°C
Impact Rating	1K08 (5J)
Vibration/Shock	ETSI 5M3
Water / Dust sealed	IP54
Certification	CE / FCC/ UL
ADA	ADA Compliant

Connectivity

The USB interface comprises an internal USB hub with connected keyboard and audio module.
This is a composite USB device and no additional drivers are required.

PC based software utility and API are available to set/control: -

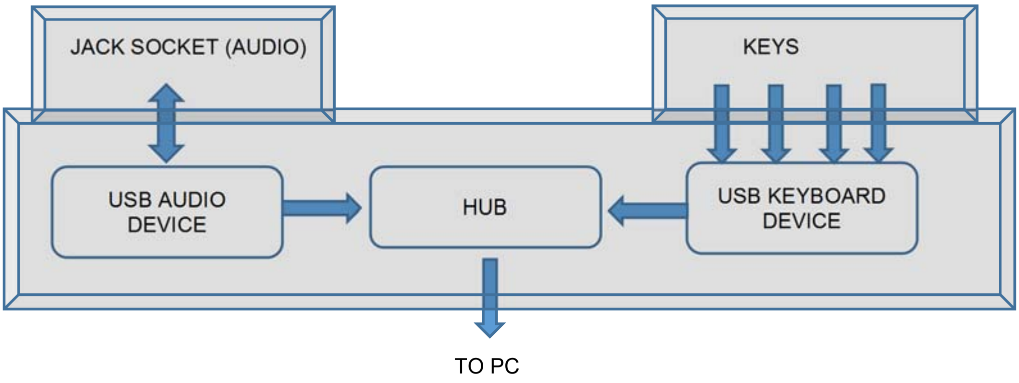
- Volume key function
- Illumination level control
- Customise the USB codes

This product is licensed under NCR's design rights, including NCR U.S. Design Patent D687,783 and European Design Registration 001887290. It incorporates proprietary technology and intellectual property retained by Keymat Technology Ltd. (trading as Storm Interface).

USB Device Information

USB HID

The USB interface comprises a USB HUB with keyboard device and audio device connected.



The following VID/PID combinations are used:

For USB HUB:	For Standard Keyboard/Composite HID/ Consumer Controlled device	For USB Audio device
• VID – 0x0424	• VID – 0x2047	• VID – 0x0D8C
• PID – 0x2512	• PID – 0x09D0	• PID – 0x0170

This document will concentrate on the Standard Keyboard/Composite HID/Consumer Controlled device. This interface will enumerate as

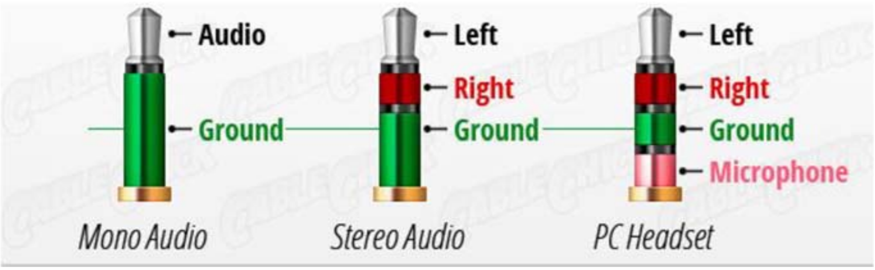
- Standard HID Keyboard
- Composite HID-datapipe Interface
- HID Consumer Controlled device

One of the advantages of using this implementation is that no drivers are required.

The data-pipe interface is used to provide the host application to facilitate customisation of the product.

Supported Audio Jack Configurations

The following jack configurations are supported.

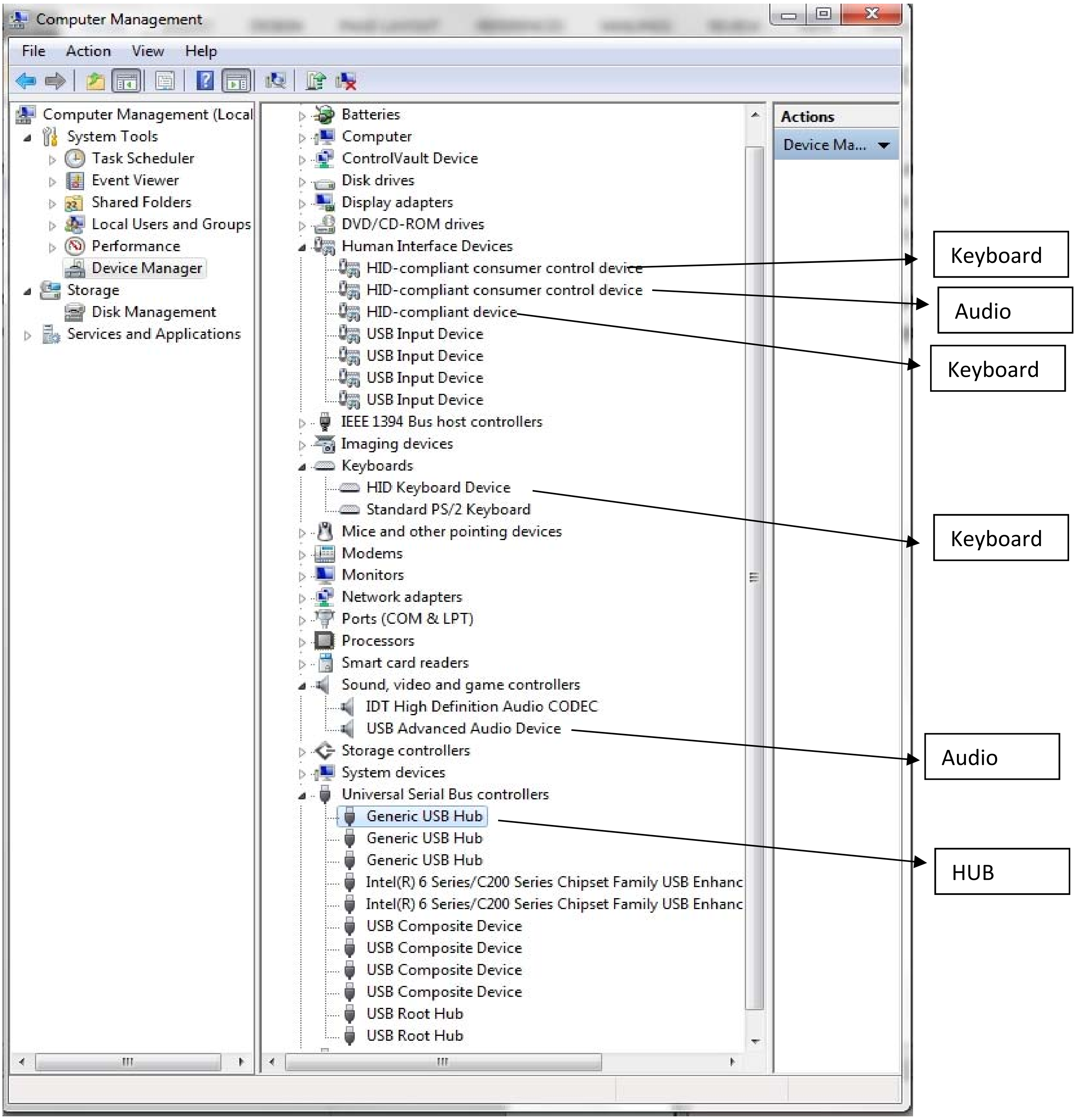


Notes: Application software should always ensure the same audio is present on both Left and Right Channels for correct mono operation. Headsets with microphones can be used but there is no microphone support.

Device Manager

When connected to a PC, the keypad should be detected by the operating system and enumerated without drivers. Windows shows following devices in the Device Manager:

(Note that other audio devices will need to be disabled in Device Manager otherwise they will take priority).

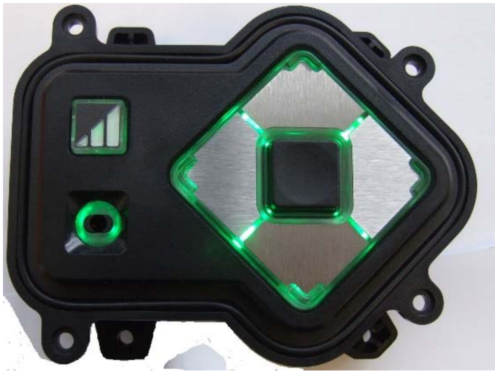


- Computer Management (Local
  - System Tools
    - Task Scheduler
    - Event Viewer
    - Shared Folders
    - Local Users and Groups
    - Performance
    - Device Manager
  - Storage
    - Disk Management
  - Services and Applications

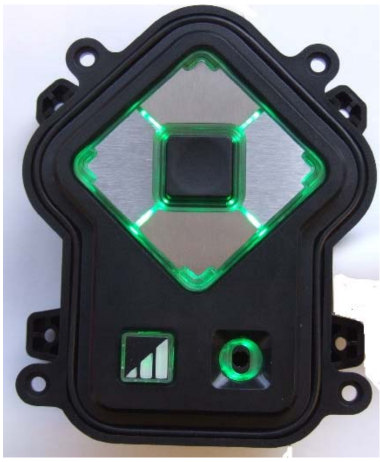
- Batteries
- Computer
- ControlVault Device
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
  - HID-compliant consumer control device
  - HID-compliant consumer control device
  - HID-compliant device
  - USB Input Device
  - USB Input Device
  - USB Input Device
  - USB Input Device
- IEEE 1394 Bus host controllers
- Imaging devices
- Keyboards
  - HID Keyboard Device
  - Standard PS/2 Keyboard
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Ports (COM & LPT)
- Processors
- Smart card readers
- Sound, video and game controllers
  - IDT High Definition Audio CODEC
  - USB Advanced Audio Device
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanc
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanc
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Composite Device
  - USB Root Hub
  - USB Root Hub

Code Tables

The default and alternate USB code tables are shown below.



Landscape



Portrait

Conventional orientation is landscape – if you move the switch to portrait mode the output codes are adjusted to suit the new orientation.

	FACTORY DEFAULT CODE TABLE			ALTERNATE CODE TABLE			CUSTOMISED CODE TABLE
	LANDSCAPE		PORTRAIT	LANDSCAPE		PORTRAIT	
Function	**Hex**	**USB**		**Hex**	**USB**		
Right	0x4F	Right Arrow	Up Arrow	0x4F	Right Arrow	Multimedia Vol Up	Set initially to the factory default values
Left	0x50	Left Arrow	Down Arrow	0x50	Left Arrow	Multimedia Vol Down	
Down	0x51	Down Arrow	Right Arrow	<0x01><0x04>	Multimedia Vol Down	Right Arrow	
Up	0x52	Up Arrow	Left Arrow	<0x01><0x02>	Multimedia Vol Up	Left Arrow	
Select	0x28	Enter	Enter	0x28	Enter	Enter	
Jack IN	0x6A	F15	F15	0x6A	F15	F15	
Jack OUT	0x6B	F16	F16	0x6B	F16	F16	
Volume	0x6C	F17	F17	0x6C	F17	F17	
Orientation Switch							
I Landscape	0x6D	F18	F18	0x6D	F18	F18	
II Portrait	0x6E	F19	F19	0x6E	F19	F19	

Using the Windows Utility to change USB Codes

If any other keypad utility software is installed (e.g EZ-Key Utility) then you should un-install that before you start.

System Requirements

The utility requires .NET framework to be installed on the PC and will communicate over the same USB connection but via the HID-HID data pipe channel, no special drivers are required.

Compatability

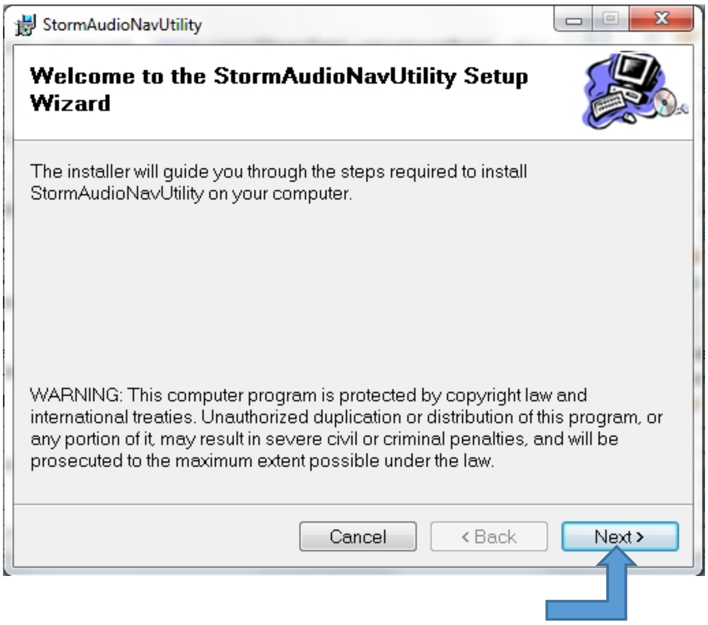
Windows 10	✓
Windows 8	✓
Windows 7	✓
Windows Vista	✓
Windows XP	Only if you install .NET framework

The utility can be used to configure the product to

- Select Code Table
- LED brightness (0 to 9)
- Test Audionav
- Create customised keypad table
- Reset to factory default
- Load Firmware

Installing the Configuration Utility

To install the Configuration Utility **doubleclick on the downloaded .exe file and the Setup Wizard will launch**

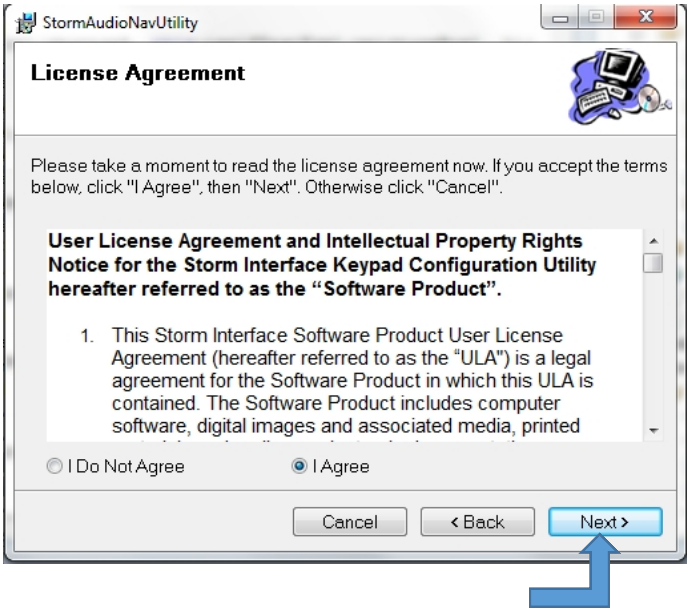

StormAudioNavUtility

Welcome to the StormAudioNavUtility Setup Wizard

The installer will guide you through the steps required to install StormAudioNavUtility on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Click on "Next"


StormAudioNavUtility

License Agreement

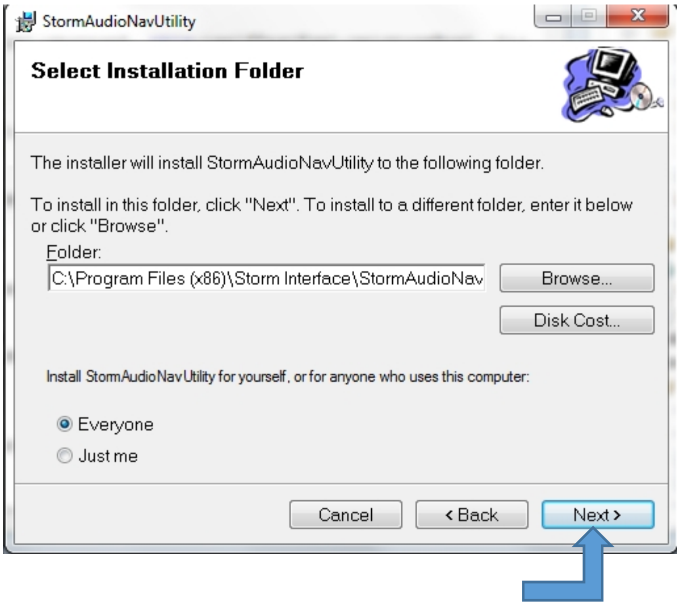
Please take a moment to read the license agreement now. If you accept the terms below, click "I Agree", then "Next". Otherwise click "Cancel".

User License Agreement and Intellectual Property Rights Notice for the Storm Interface Keypad Configuration Utility hereafter referred to as the "Software Product".

1. This Storm Interface Software Product User License Agreement (hereafter referred to as the "ULA") is a legal agreement for the Software Product in which this ULA is contained. The Software Product includes computer software, digital images and associated media, printed

○ I Do Not Agree ● I Agree

Select "I Agree" and Click on "Next"


StormAudioNavUtility

Select Installation Folder

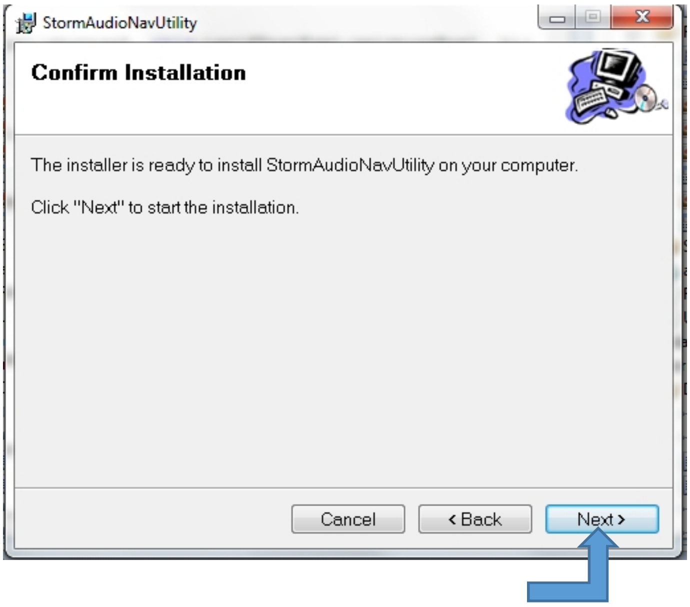
The installer will install StormAudioNavUtility to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files (x86)\Storm Interface\StormAudioNav

Install StormAudioNavUtility for yourself, or for anyone who uses this computer:

● Everyone
○ Just me


StormAudioNavUtility

Confirm Installation

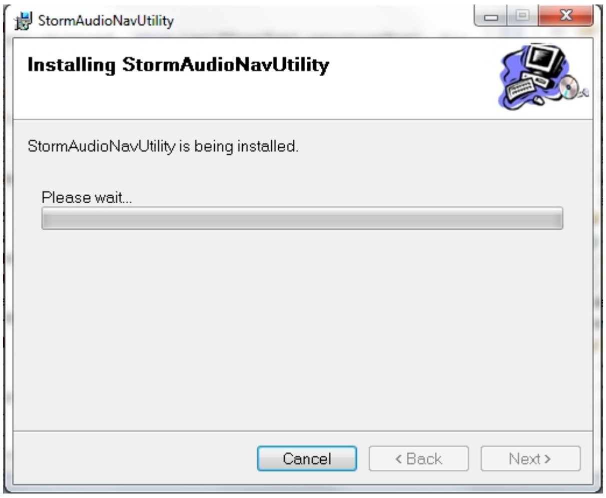
The installer is ready to install StormAudioNavUtility on your computer.

Click "Next" to start the installation.

Select if you would like to install for just you or everyone and select location if you do not want to install at default location. Then click on "Next"

Click on "Next" to confirm.

The "Disk Cost" shows available space at your chosen folder. The program requires 10MB of space.

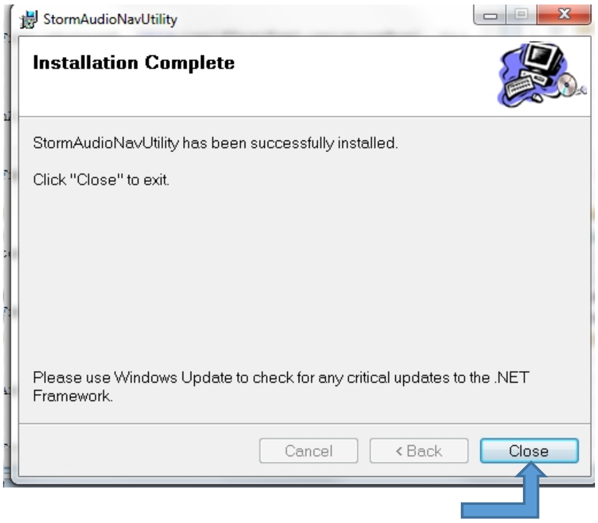


StormAudioNavUtility

Installing StormAudioNavUtility

StormAudioNavUtility is being installed.

Please wait...



StormAudioNavUtility

Installation Complete

StormAudioNavUtility has been successfully installed.

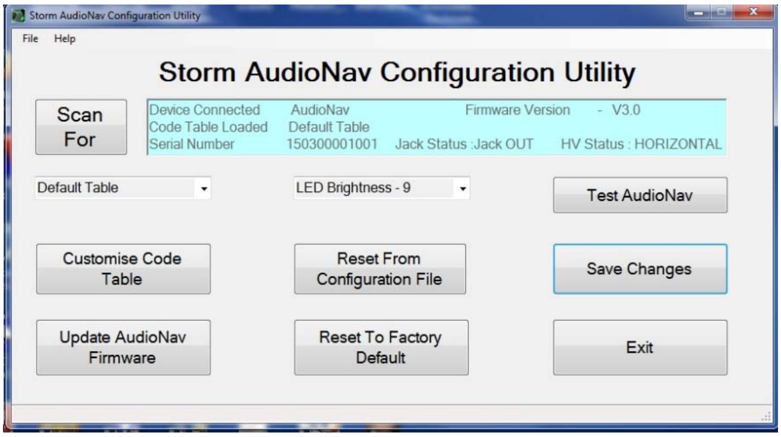
Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Click on “Close” for successful installation.

A shortcut will be installed on Desktop.

Double-click this to start the Utility and the following screen will appear. If an audio-nav is connected it will be detected automatically and the details displayed.



Storm AudioNav Configuration Utility

Storm AudioNav Configuration Utility

Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Default Table
Serial Number 150300001001 Jack Status :Jack OUT HV Status : HORIZONTAL

Default Table

LED Brightness - 9

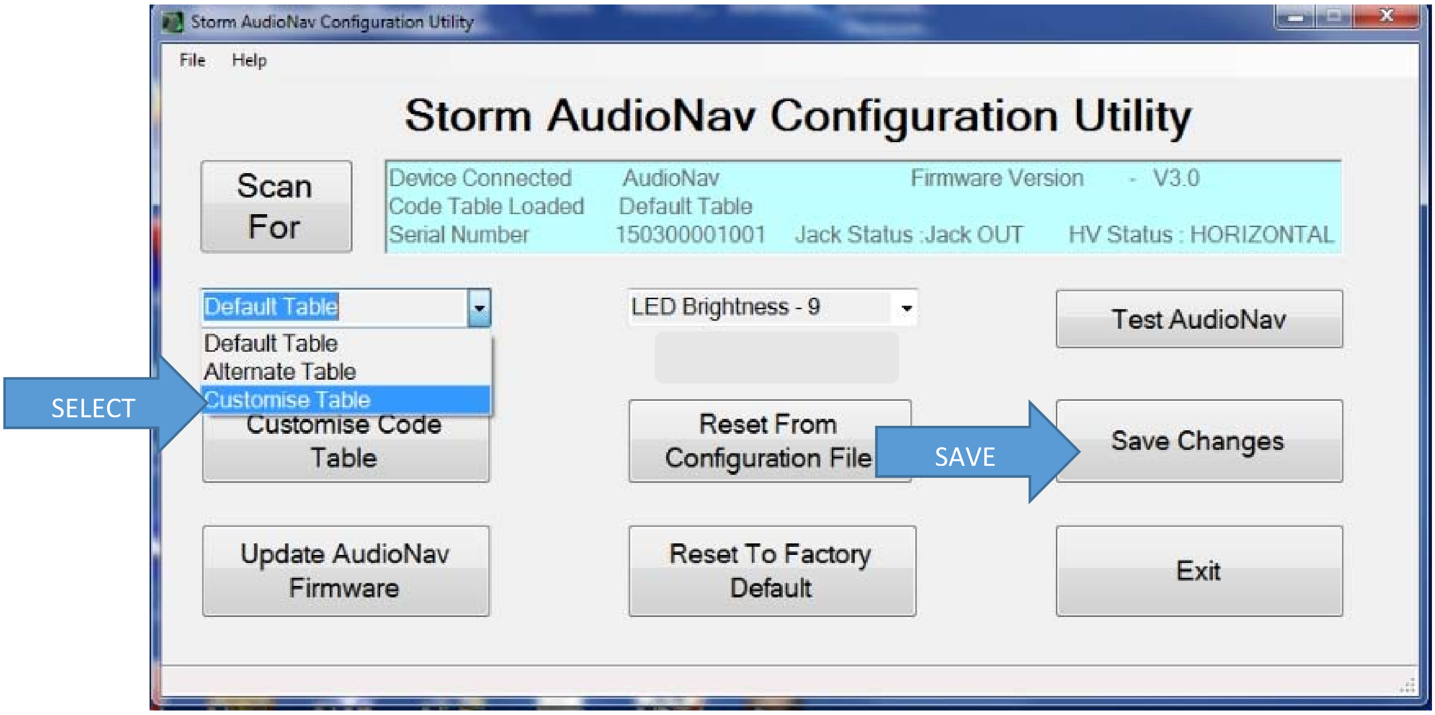
Select Code Table

The user can select from three tables:

	FACTORY DEFAULT OUTPUT CODE TABLE		ALTERNATE CODE TABLE		CUSTOMISED CODE TABLE
Function	**Hex**	**USB Description**	**Hex**	**Description**	
Right	0x4F	Right Arrow	0x4F	Right Arrow	Set initially to the factory default values
Left	0x50	Left Arrow	0x50	Left Arrow	
Down	0x51	Down Arrow	<0x01><0x04>	Multimedia Vol Down	
Up	0x52	Up Arrow	<0x01><0x02>	Multimedia Vol Up	
Select	0x28	Enter	0x28	Enter	
Jack IN	0x6A	F15	0x6A	F15	
Jack OUT	0x6B	F16	0x6B	F16	
Volume	0x6C	F17	0x6C	F17	
Orientation Switch					
Landscape	0x6D	F18	0x6D	F18	
Portrait	0x6E	F19	0x6E	F19	

Once a table has been selected then the keypad will hold that configuration unless it is disconnected.

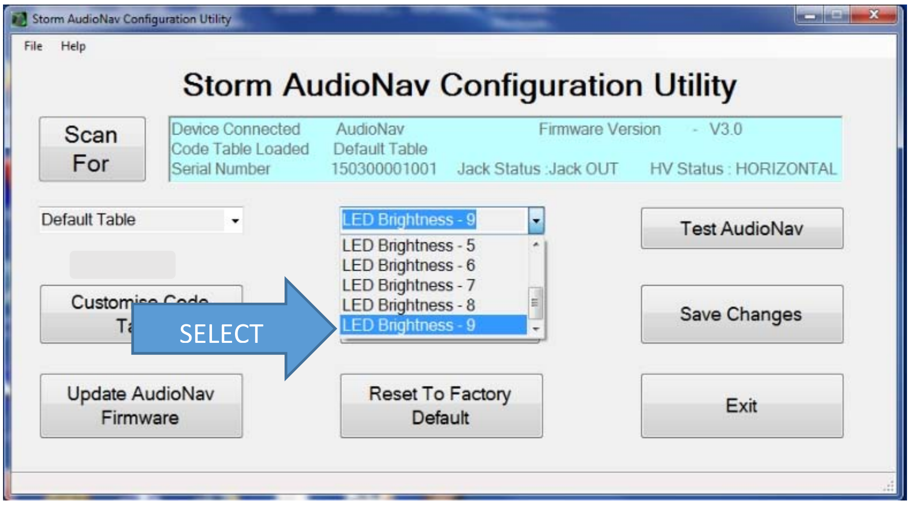
Once the keypad has been disconnected that configuration will be lost unless you save the configuration in memory by clicking on “Save Changes”



Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Default Table
Serial Number 150300001001 Jack Status :Jack OUT HV Status : HORIZONTAL

LED Brightness

This will set the brightness of the LEDs. The selection is from 0 to 9.



Storm AudioNav Configuration Utility

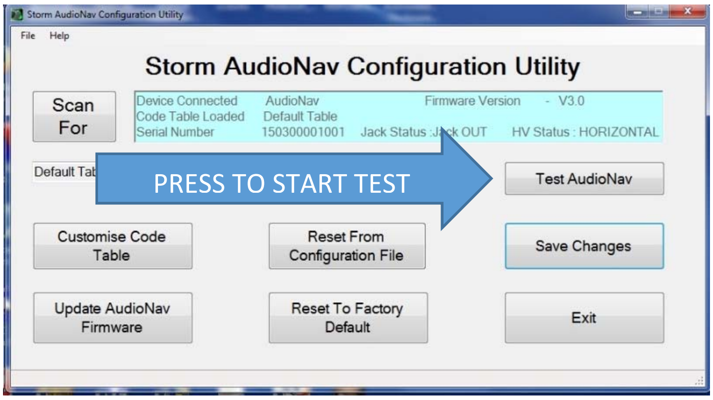
Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Default Table
Serial Number 150300001001 Jack Status :Jack OUT HV Status : HORIZONTAL

LED Brightness - 9

Test AudioNav

This will test all the functionality of AudioNav keypad.

- illumination dimming levels
- Key test
- Jack in/out
- H/V Switch
- Audio test

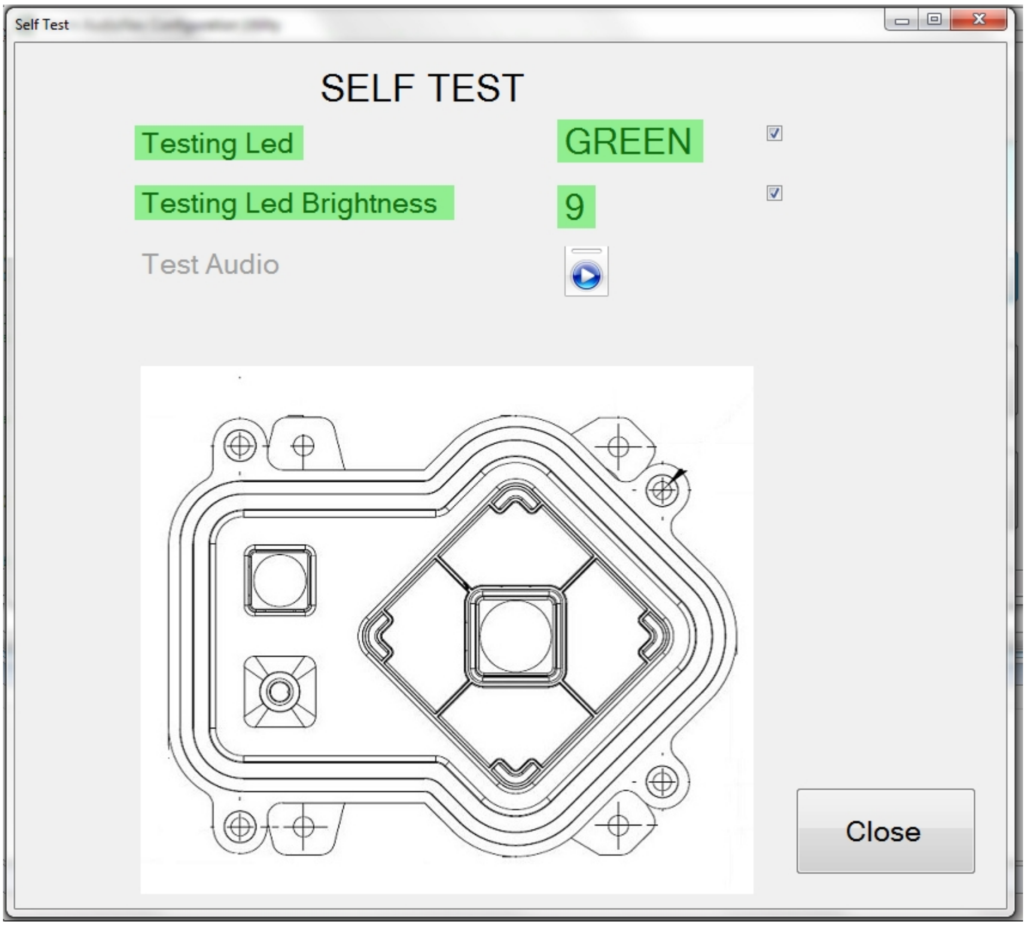

Storm AudioNav Configuration Utility

Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Default Table
Serial Number 150300001001 Jack Status :Jack OUT HV Status : HORIZONTAL

PRESS TO START TEST

To test the audio of the AudioNav, please make sure that this is the default device.

After the audio test press each key on keypad, the relevant detected key will be shown.


SELF TEST

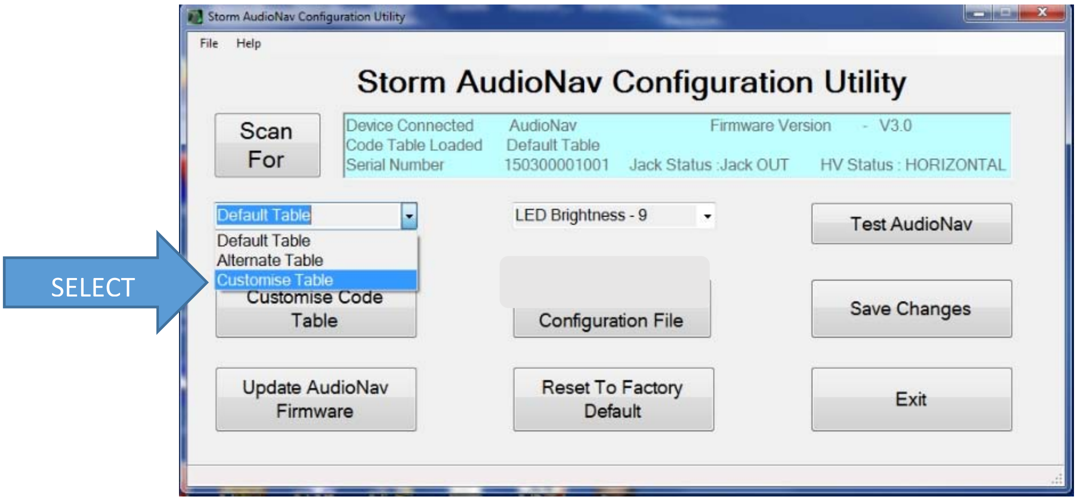
Testing Led GREEN ☑

Testing Led Brightness 9 ☑

Test Audio

Press close when finished.

Customise Code Table



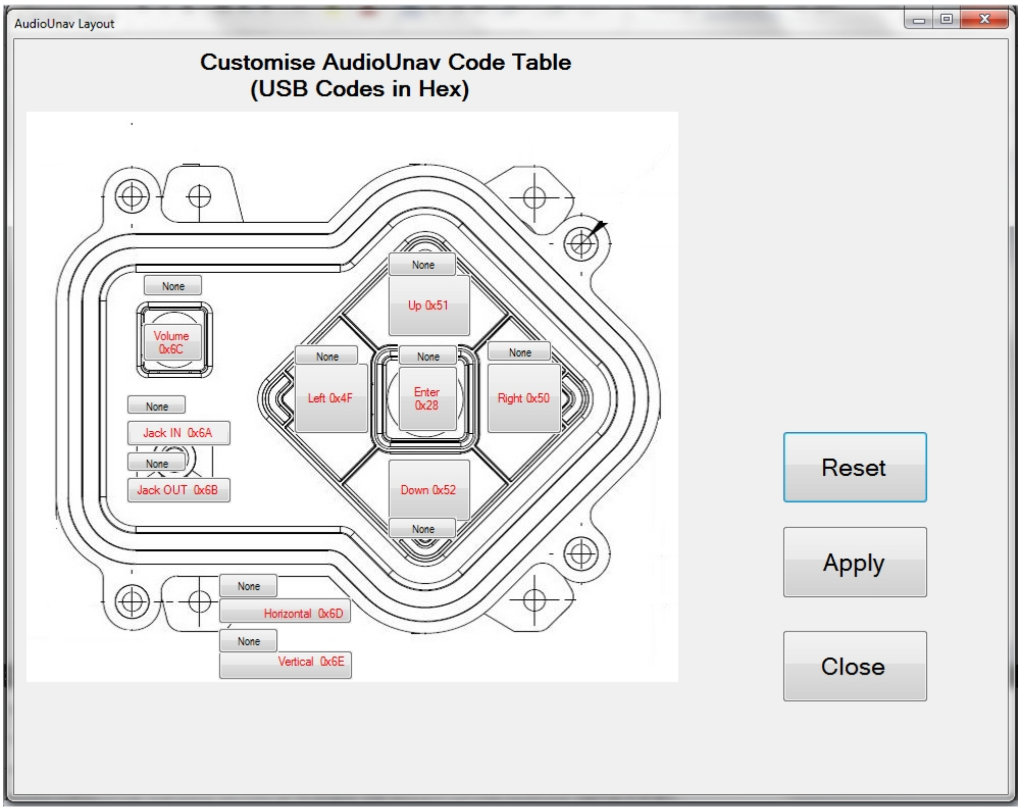
Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Default Table
Serial Number 150300001001 Jack Status :Jack OUT HV Status : HORIZONTAL

User can only enter into this menu if the AudioNav customise table is selected.

Note that Multimedia Control Codes (Vol Up / Down) are not available in Customised Table.

The following will be displayed when “Customise code” is clicked.

Please note: JACK IN/OUT and Horizontal/Vertical codes can also be customised.

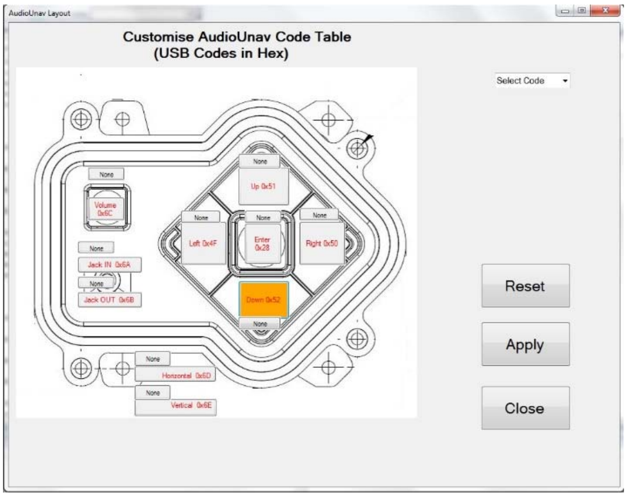


The content of this communication and/or document, including but not limited to images, specifications, designs, concepts and information is confidential and is not to be used for any purpose or disclosed to a third party without the express and written consent of Keymat Technology Ltd., Copyright 2018. All rights reserved.

Choosing a USB Code

The current customised code table will be displayed from memory on the keypad.
Attached to each key is another button ("NONE"), this shows the modifier for each key.

To customize a key, click on the key and
Key Code combo box will appear, with "Select Code"
.
The button colour will change to "Orange"

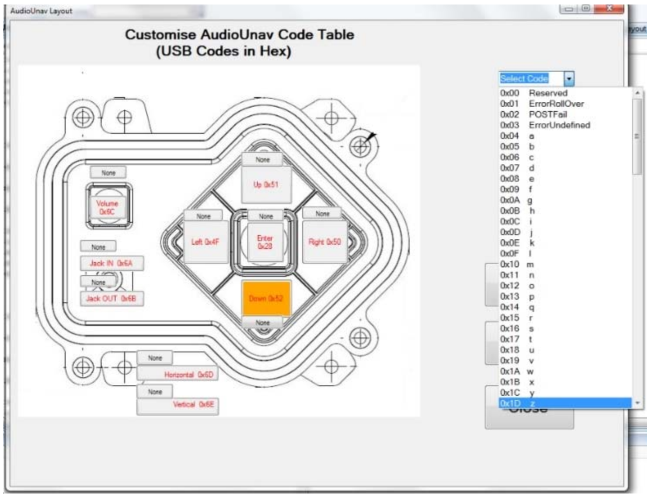

AudioUnav Layout

Customise AudioUnav Code Table
(USB Codes in Hex)

Now press on the down arrow on the combo box:
This will display all the codes that can be selected.

These codes are the ones defined by USB.org.

Once code is selected, the code will be displayed on the selected button.

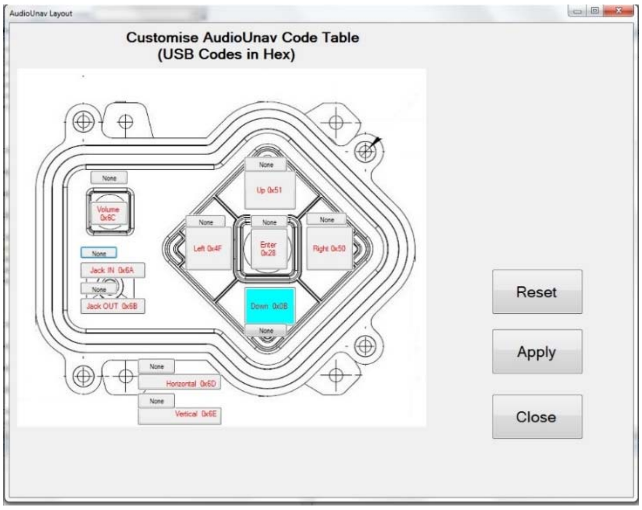

AudioUnav Layout

Customise AudioUnav Code Table
(USB Codes in Hex)

Select Code

0x00	Reserved
0x01	ErrorRollOver
0x02	POSTFail
0x03	ErrorUndefined
0x04	a
0x05	b
0x06	c
0x07	d
0x08	e
0x09	f
0x0A	g
0x0B	h
0x0C	i
0x0D	j
0x0E	k
0x0F	l
0x10	m
0x11	n
0x12	o
0x13	p
0x14	q
0x15	r
0x16	s
0x17	t
0x18	u
0x19	v
0x1A	w
0x1B	x
0x1C	y
0x1D	z

In this example I have selected "e" and code is represented by 0x08 and button colour will change to Aqua.


AudioUnav Layout

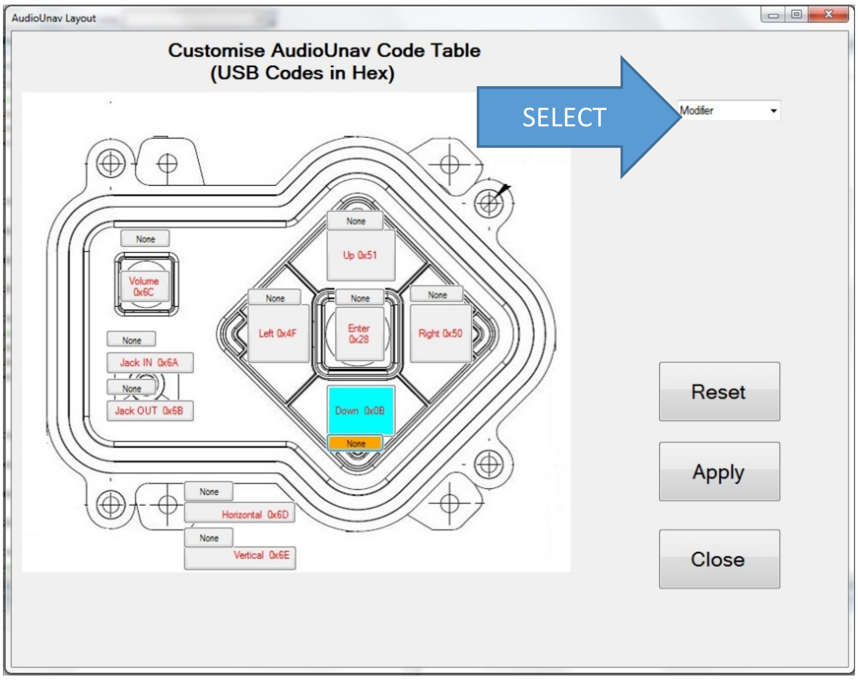
Customise AudioUnav Code Table
(USB Codes in Hex)

Press the "Apply" button and the code will be sent to the AUDIONAV.

When you press key "Down" on keypad,
"e" will be sent to the relevant application.

Adding a modifier

Now if you wanted a “E” (uppercase) then you need to add a SHIFT modifier for that key. Click on the modifier button for that key.



The background colour for modifier button will change to orange and modifier combo box will appear. Select down arrow key on modifier combo box and the following selection will be available:

NONE

L SHT – Left Shift

L ALT – Left Alt

L CTL – Left Ctrl

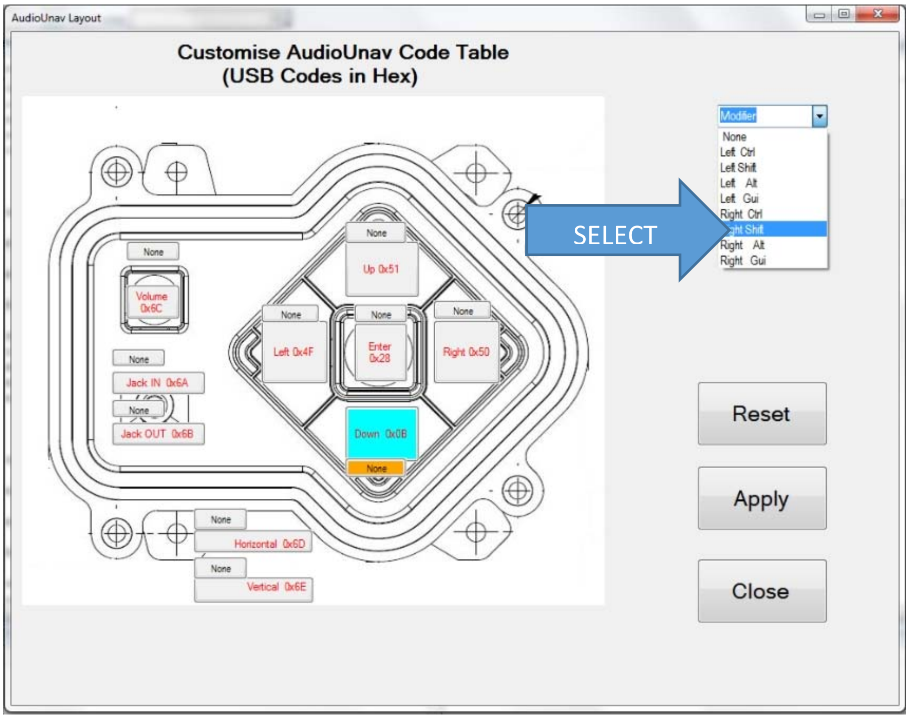
L GUI – Left Gui

R SHT – Right Shift

R ALT – Right Alt

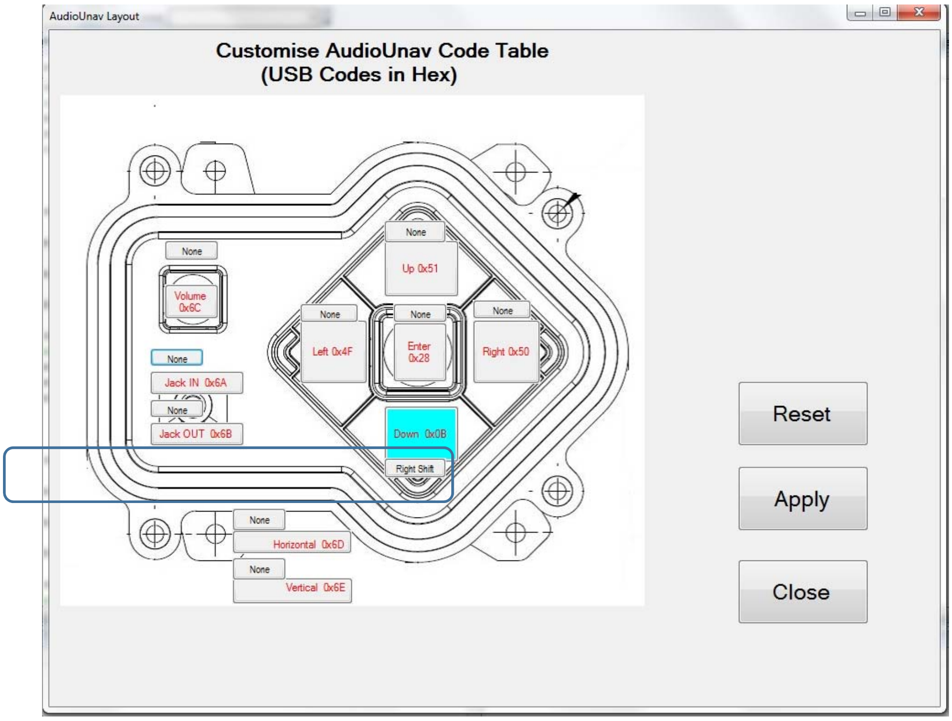
R CTL – Right Ctrl

R GUI – Right Gui



Select either L SHT or R SHT – I have selected R SHT.

The R SHT modifier is now displayed on button and background colour changed to grey. Now if you click on “Apply” and if successfully transferred then pressing “down” on keypad will display “E” (uppercase).

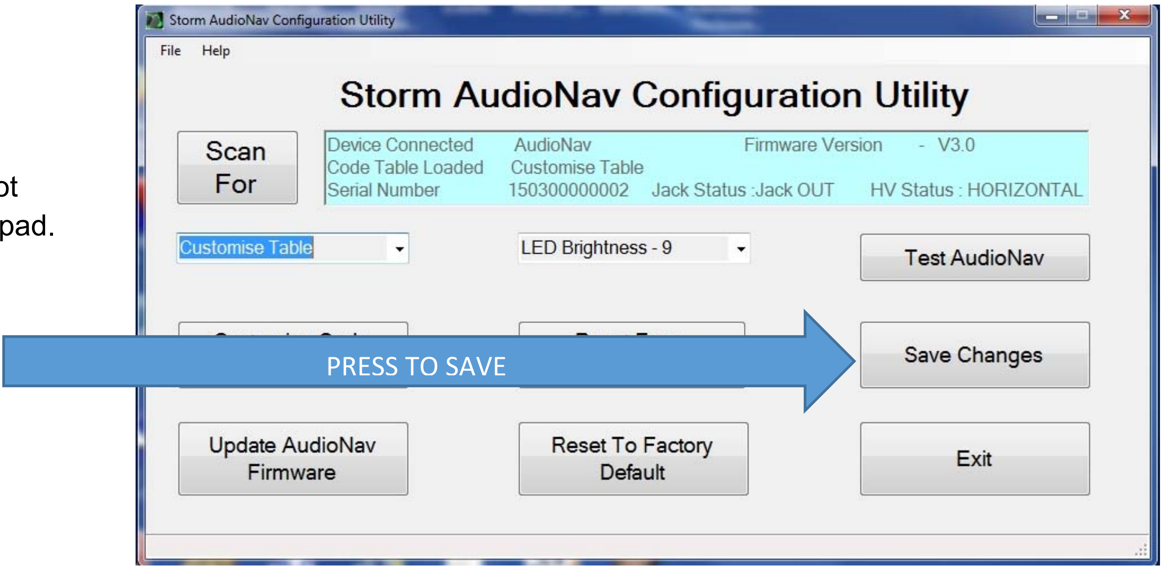


If you did not want the current setting then click on “Reset” then all buttons will revert to original coding and then click on “apply” to send this coding to AudioNav keypad.

“Close” will exit the customize form and return back to main screen.

Saving Changes

If you don't press
"Save Changes"
your changes will not
be saved to the keypad.



Storm AudioNav Configuration Utility

Device Connected AudioNav Firmware Version - V3.0
Code Table Loaded Customise Table
Serial Number 150300000002 Jack Status :Jack OUT HV Status : HORIZONTAL

Factory Default

Clicking on "Factory Default" will reset the keypad to factory settings

Code Table – Default

LED brightness – 9

API for controlling the AudioNav device from the Host Computer

This section provides details on how the AudioNav can be controlled from a host that has USB capabilities.

List of Messages

(Structure of Messages from Host to AudioNav™ is on the following pages

ID	Name	Description
01	Device Status Request	Output the firmware version & selected parameters
02	LED Brightness	Adjust led brightness.
03	Reserved	Reserved for future use
04	Reserved	Reserved for future use
05	Load New code table	Load new code table
06	Reserved	Reserved for future use
07	Keypad Type	Select layout table
08	Reserved	Reserved
09	Write to default	AudioNav writes configuration data from ram to flash
10	Reset to factory default	Reset device back to factory default
11	Reserved	Reserved for future use
12	Load Firmware	Sets the AudioNav to detect the device loader for firmware loading
13	Reserved	Reserved for future use
14	Set Serial Number	Write 12 digit serial number

Structure of Messages from AudioNav to Host

01 Key Press Code	sends a key scan code back to HOST when a key is pressed on keypad

AudioNav Device Communications

AudioNav keypad uses the ASCII/binary Message format described below. Every message that is sent from a host should be acknowledged with the control byte ACK (0x06). A retransmission should be initiated if an NAK (0x15) is received or if no acknowledge is received at all.

Message Formats

A	Alpha character, 'A'-'Z' and 'a' - 'z'
C	Control character one byte in length.
H	Hexadecimal characters, ‘0’-‘9’, ‘A’-‘F’
N	Numeric character, ‘0’-‘9’
S	Special characters, entire character set 0x00 - 0xFF

ASCII Message Format

	Message Field	Type	Length	Description
1	STX	C	1	Control character Start of Text = 0x02
2	Message Id	H	2	Defines the type of message and format of the data field
3	Data Length	H	2	Hexadecimal value represented in ASCII defines the number of bytes in the data field. '00' to 'FF'. Maximum data field size is 256 bytes.
4	Data Field	S	var	In binary format
5	ETX	C	1	Control character ETX = 0x03
6	LRC	C	1	Longitudinal Redundancy Check Digit, calculated on all previous data including STX

Message ID Definitions

Here is a general table describing the message Ids, more detailed descriptions for each message Id follows. When a message is one way only, the Message Id. is the same for both the message and response.

ID.	Message	Description
01	Device Status Request	Host To AUDIONAV keypad – Output the firmware version and all currently selected parameters
02	LED Brightness	Host To AUDIONAV keypad – adjust led brightness. (default: 0)
03	Reserved	RESERVED
04	Reserved	RESERVED
05	Load New code table	Host To AUDIONAV keypad – Load new code table
06	Reserved	RESERVED
07	Keypad Table	Host To AUDIONAV keypad – Select layout table 0 – Default Table 1 – Alternate Table 2 – Customised
08	Reserved	Reserved
09	Write to default	Host To AUDIONAV – AudioNav writes configuration data from ram to flash.
10	Reset to factory default	Host To AUDIONAV – Reset device back to factory default
11	Reserved	RESERVED
12	Load Firmware	Host To AUDIONAV– Sets the AudioNav to detect the device loader for firmware loading
13	Reserved	RESERVED
14	Set Serial Number	Host to AUDIONAV– to store a serial number (12 digits)

Error Code

Every response message contains one of the following error codes:

00	No error
01	Command not recognized
02	Command not support at this stage
03	Parameter not supported
04	Hardware fault

Device Status (01)

Host sends this message to AudioNav to request the status of the AudioNav keypad

Host Device		AUDIONAV-
[01]	⇒	
	⇐	[01][ec][lb][kt][Js][Hs][20*kc][fw][sn]

AudioNav **Status Response**

Keypad sends this message to Host in response to the Device Status message.

	Data Field	Type	Length	Description
ec	Error Code	SH	2	
Lb	LED Brightness	SN	1	Value (0 – 9)
Kt	Keypad Table	SN	1	0 – Default Table 1 – Alternate Table 2 – Customised Table
Js	Jack status	SN	1	0 – Jack IN, 1 – Jack Out
Hs	Horizontal/Verticle	SN	1	0 – Vertical 1 - Horizontal
Kc	Keycode	SH	20	Customised keycode for each key
fw	Firmware Version	ANS	20	Left justified, if Firmware Version is less than 20 then just add enough spaces after the Firmware Version until this field is completed, for instance, “123456” becomes: “123456 “
sn	Serial Number	ANS	12	Returns serial number YYQQXXXXXXXX Where YY – year, QQ – Quarter XXXXXXXX – Sequential number

LED Brightness Command (02)

Host sends this message to control brightness of LEDs

	Data Field	Type	Length	Description
1	LED brightness	SN	1	0 - 9

LED Brightness Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AudioNav
[02][lb]	⇨	
	⇦	[02][ec]

Note: LED brightness of 0 value indicates LEDs are off

LED brightness of 9 value indicates full brightness

Reserved (03)

Reserved (04)

Load New Key Code Table Command (05)

Host sends this message to Load New Code Table

	Data Field	Type	Length	Description
1	Load New Code Table	SH	20	Key Code Table:

Load New Table Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AudioNav
[05][lt][20 scan codes]	⇨	
	⇦	[05][ec]

Note: Length is always 20,

Format of table is as follows:

<modifier for key 1><code for Key 1><modifier for key 2><Code for Key 2>..........etc

The code table is specified in the user manual together with the modifier code. For example to program the following for 4 way :

Key 1 – A

Key 2 – a

Key 3 – 9

Key 4 - !

<0xE1><0x04><0x00><0x04><0x00><0x26><0xE5><0x1E>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>< 0x00><0x00>

Note: 20 bytes must be sent, for unused key code pad the values with 0x00.

Note: For shift modifiers there is a left and right modifiers value defined. So we can use 0xE1 – Left Shift and 0xE5 – Right shift. Similarly there is left and right Alt

Reserved (06)

Keypad Table Command (07)

Host sends this message to set keypad type

	Data Field	Type	Length	Description	
1	Keypad Type	SN	1	0 – Default Table 1 – Alternate Table 2 – Customised Table	

Keypad Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AudioNav
[07][bp]	⇨	
	⇦	[07][ec]

Reserved (08)

Write Config Data To Flash command (09)

Host sends this command to request the AUDIONAV to write the configuration data from RAM to FLASH. This command has no data associated with it.

Host Device		AUDIONAV-
[09]	⇨	
	⇦	[09][ec]

RAM to FLASH command **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Reset To Factory Default command (10)

Host sends this command to request the AUDIONAV to reset parameters back to factory default. This command has no data associated with it.

Reset To Factory Default **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AUDIONAV-
[10]	⇒	
	⇐	[10][ec]

Reserved (11)

Enable BSL Command (12)

Host sends this command to request the AUDIONAV to start downloader

Enable BSL command **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AUDIONAV-
[12]	⇒	
	⇐	[12][ec]

Reserved (13)

Set Serial Number command (14)

Host sends this command to set the serial number of the device in format YYQQXXXXXXXX

	Data Field	Type	Length	Description
1	Serial Number	ANS	12	YYQQXXXXXXXX

Set Serial Number command **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		AUDIONAV-
[14][SN]	⇨	
	⇦	[14][ec]

(01) Key Press Code

With the USB stack configured for a standard keyboard interface, the AudioNav sends appropriate key report to HOST when a key is pressed on keypad.

Keyboard Report

HID Keyboard Report Format

	BIT7	***BIT6***	***BIT5***	***BIT4***	***BIT3***	***BIT2***	***BIT1***	***BIT0***
Byte0	*Right GUI*	*Right Alt*	*Right Sft*	Right Ctrl	Left GUI	Left Alt	Left Shift	Left Ctrl
Byte1	*Reserved*							
Byte2	Key_array[0]							
Byte3	Key_array[1]							
Byte4	Key_array[2]							
Byte5	Key_array[3]							
Byte6	Key_array[4]							
Byte7	Key_array[5]							

For example if user has configured for Default Table. If the user now presses the top key, which is “<<” and USB code of 72. Then keyboard report sent to host would be:

Byte 0 – 0

Byte 1 – 0

Byte 2 – 72

Byte 3 – 0

Byte 4 – 0

Byte 5 – 0

Byte 6 – 0

Byte 7 – 0

Now if the user customizes the top key to be “R SHIFT” (modifier) and USB code for “a” (04). If the user presses the top key, then the keyboard report sent to host would be:

Byte 0 – 20 This is Right Shift modifier.

Byte 1 – 0

Byte 2 – 04

Byte 3 – 0

Byte 4 – 0

Byte 5 – 0

Byte 6 – 0

Byte 7 – 0

Host API Library - Overview

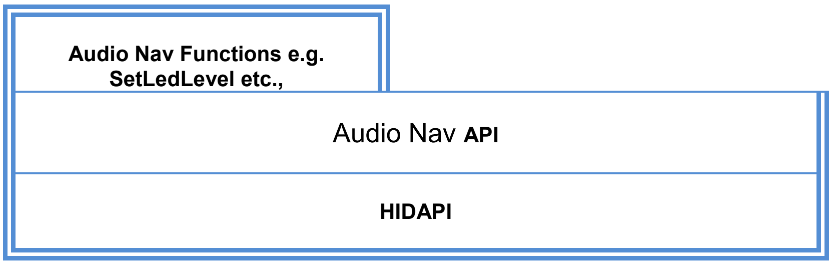
The Host API Library for the AudioNav is a middleware application between the Host application and Audio Nav device. You can download this together with the HIDAPI library from www.storm-interface.com.

- Audio Nav API – The AudioNavApi library allows for the host application to invoke Audio Nav functions as listed above. The API encapsulates all the communications to USB and provides a simple API for the host application developers.

- HIDAPI - This is a third party library, which allows an application to interface with USB HID-Compliant devices on Windows, Linux, and Mac OS X. While it can be used to communicate with standard HID devices like keyboards, mice, and Joysticks, it is most useful with custom (Vendor-Defined) HID devices. This allows for host software to scan for the device using its VID/PID.

The Audio Nav uses USB for communicating with the host. It includes an HID-compliant device . One of the advantages of using this implementation, which uses only HID interfaces, is that no drivers are required on host system.

The protocol for communicating with host is described fully in the following pages. The basic architecture is shown below.



The developer does not need to worry about the communication at low level. You can request source code from us for the library so it can be ported to your specific platform. Currently the library has been tested on Windows and Linux (Ubuntu) platform.

The API makes the following functions available to developers

Page

Message Types

This is referenced in below functions:

```
    enum REQUEST_TYPE{      // message types

    DEVICE_STATUS = 1,              ///Device status message
    LED_BRIGHTNESS,                 //< set led brightness
    RESERVED_1,                           ///MID_RESERVED_6
    RESERVED_2,                     // MID_RESERVED_6
    LOAD_NEW_TABLE,                       //load new key code table
    RESERVED_3,              // MID_RESERVED_6
    KEYPAD_TYPE,                    // set keypad type 0 - default table, 1 -
alternate 2- customise
    RESERVED_4,                           //MID_RESERVED_6
    WRITE_DEFAULT,                      // Write defaults values from ram to flash
    RESERVED_FO_FACTORY_DEFAULT,    // reset the setting to factory default
    RESERVED_5,                     //MID_RESERVED_6
    ENABLE_BSL,                     //start downloader
    RESERVED_6                //MID_RESERVED_6

      }
```

InitialiseStormUSBDevice

This function is used to initialise the Audio Nav. The Audio Nav is identified by the Product PID and Manufacturer VID. This are assigned to Keymat:

- Vendor ID – 0x2047
- Product ID – 0x09D0

On successful finding the Audio Nav the manufacturer_local will be filled with "Storm Interface" and product_local will be filled with "AUDIO NAV". If not successful both of the strings will be filled with "none"

Parameters :

storm_vid	-	Vendor ID
product_pid	-	Product ID
manufacturer	-	vendors name will be stored
product	-	product name will be stored

Return Value:

True for success
False for failure.

```
///\brief InitializeStormUSBDevice is called at the beginning of the
application to

///Setup the PRODUCT ID (PID) and product vid

///\return false on failure, true on success.

///On failure, call GetErrorCode() to retrieve the error

///

bool InitializeStormUSBDevice( int storm_vid, int product_pid);
```

GetDeviceStatus

This function retrieves status information about the Audio Nav. For example, Jack status, HV switch status, led level status etc. All information is stored in DEVICE_INFO structure.

Parameters :

```
typedef struct
{
      unsigned char           led_brightness;
      unsigned char           keypad_table;
      unsigned char           jack_status;
      unsigned char           HV_status;
      unsigned char           keyCode[20]; //currently keytable in use
      std::string             version;
      std::string             serialNumber;
} DEVICE_INFO;
```

_deviceInfo - DEVICE_INFO sturcture, that will be filled by the function
timeToWait - maximum time to wait for command to complete

Return Value:

True for success
False for failure.

```
///\brief GetDeviceStatus Retrieves the USB Display’s status information including:
jack status, HV switch status, Firmware Name.
///The data are returned in  a DEVICE_INFO structure
///\param _deviceInfo is a pointer to a DEVICE_INFO structure that receives
information retrieved from the Audio Nav.
///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
///\return 0 on success, negative error code on failure
///

Int GetDeviceStatus( DEVICE_INFO *_deviceInfo, int _timeToWait );
```

SetLedLevel

This function sets the brightness of the led. The led level can be set with values 0 to 9.

Parameters :

Int ledLevel
timeToWait - maximum time to wait for command to complete

Return Value:

0 for success

```
      ///\brief SetLedLevel This function sets led brightness level from 0 to 9,
where 0 is off
      ///                             and 9 is on.
      ///\param ledLevel    used to set led level
      ///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
      ///\return 0 on success, negative error code on failure
      ///     Possible error codes are:
      ///         DEVICE_INFO_STRUCTURE_NULL          = User app passed in NULL
pointer for DEVICE_INFO structure
      ///         NO_USB_DISPLAY_CONNECTED                        = No keypad is
connected so cannot retrieve info
      ///         REQUEST_TIMEOUT                           = Could not retrieve the
info in the time alloted.
      ///
      DLLDEF int                                SetLedLevel( int ledLevel, int
_timeToWait );
```

LoadCodeTable

This function loads the keycode table in customise table.

Parameters :

Int *keyCodePtr – pointer to code table must hold 20 values including modifier. For each key the values must be [modifier, USB key code].
timeToWait - maximum time to wait for command to complete

Return Value:

0 for success

```
      ///\brief LoadCodeTable This function loads a new code table to customise
table in AudioNav
      ///
      ///\param KeyCodePtr - Point to new code table
      /// param keyCodeLen - length of keycode - Must be 20.
      ///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
      ///\return 0 on success, negative error code on failure
      ///     Possible error codes are:
      ///         DEVICE_INFO_STRUCTURE_NULL          = User app passed in NULL
pointer for DEVICE_INFO structure
      ///         NO_USB_DISPLAY_CONNECTED                        = No keypad is
connected so cannot retrieve info
      ///         REQUEST_TIMEOUT                           = Could not retrieve the
info in the time alloted.
      ///
      DLLDEF int                                LoadCodeTable( char *keyCodePtr, int
keyCodeLen, int _timeToWait );
```

SetKeypadTable

This function sets the current keypad table that will be used. 0 – default, 1 – alternate, 2 - customise

Parameters :

Int KeypadTable 0 – default, 1 – alternate, 2 - customise
timeToWait - maximum time to wait for command to complete

Return Value:

0 for success

```
            ///\brief SetKeypadTable This function sets which table is currently
used.
      ///
      ///\param KeyCodeTable - 0 - default, 1 - alternate 2- customise

      ///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
      ///\return 0 on success, negative error code on failure
      ///     Possible error codes are:
      ///         DEVICE_INFO_STRUCTURE_NULL          = User app passed in NULL
pointer for DEVICE_INFO structure
      ///         NO_USB_DISPLAY_CONNECTED                        = No keypad is
connected so cannot retrieve info
      ///         REQUEST_TIMEOUT                           = Could not retrieve the
info in the time alloted.
      ///
      DLLDEF int                                  SetKeypadTable(int keyCodeTable, int
_timeToWait );
```

WriteDefaultToFlash

This function commnds the AudioNav to commit current values to flash.

Parameters :

timeToWait - maximum time to wait for command to complete

Return Value:

0 for success

```
            ///\brief WriteDefaultToFlash This function writes changed values to
Flash
      ///
      ///\param None

      ///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
      ///\return 0 on success, negative error code on failure
      ///     Possible error codes are:
      ///         DEVICE_INFO_STRUCTURE_NULL          = User app passed in NULL
pointer for DEVICE_INFO structure
      ///         NO_USB_DISPLAY_CONNECTED                        = No keypad is
connected so cannot retrieve info
      ///         REQUEST_TIMEOUT                           = Could not retrieve the
info in the time alloted.
      ///
      DLLDEF int                                WriteDefaultToFlash(int _timeToWait
);
```

ResetToFactoryDefault

This function commnds the AudioNav to reset the Audio Nav to factory default.

Parameters :

timeToWait - maximum time to wait for command to complete

Return Value:

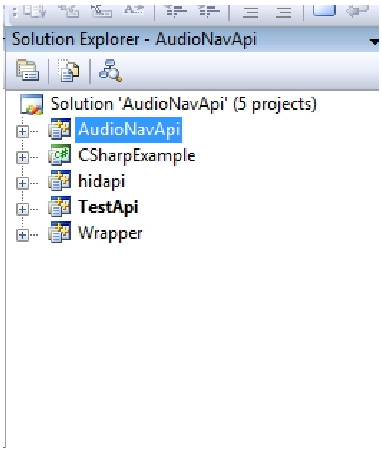
0 for success

```
            ///\brief ResetToFactoryDefault This function reset AudioNav to factory
default
      ///
      ///\param None

      ///\param _timeToWait is the time in milliseconds to wait for the data to be
retrieved.
      ///\return 0 on success, negative error code on failure
      ///     Possible error codes are:
      ///         DEVICE_INFO_STRUCTURE_NULL          = User app passed in NULL
pointer for DEVICE_INFO structure
      ///         NO_USB_DISPLAY_CONNECTED                        = No keypad is
connected so cannot retrieve info
      ///         REQUEST_TIMEOUT                           = Could not retrieve the
info in the time alloted.
      ///
      DLLDEF int                                ResetToFactoryDefault(int
_timeToWait );
```

Workspace

Click on AudioNavApi (microsoft visual studio solution) and the workspace will be launched. The workspace has 5 projects:



Solution Explorer - AudioNavApi

Solution 'AudioNavApi' (5 projects)
- AudioNavApi
- CSharpExample
- hidapi
- TestApi
- Wrapper

AudioNavApi – This is the API as described above.

Hidapi - This is freeware api for low level communication with USB device.

TestApi - This shows how the AudioNavApi is used using c++.

Wrapper - This wrapper is created to allow to integrate the AudioNavApi to languages such as c sharp

CSharpExample - Show how the AudioNavApi + Wrapper is used to communicate with the device.

Remote Update of Device Firmware

This is to allow customers to check firmware version

or remotely update the firmware

in products that are already installed.

Files included

- BSL430.dll
- AudionavApi.dll
- AudionavDownloaderUtility.exe

Program Usage

The utility will work on any windows platform, and allows you to update an AudioNav with a new version of firmware.

In operation it will

- Connect to the AudioNav
- Save the AudioNav existing configuration data, including serial number, keycodes.
- Update the AudioNav with the new firmware.
- Restore the AudioNav stored configuration data, including serial number

Run the following command in a batch file

```
AudioNavDownloaderUtility -p AUDIONAV -f FILENAME -r NUMBER
```

where :-

FILENAME is a text file which is the firmware file (e.g. 000-IC-169-EZKV05-DWG.txt)

NUMBER – (best value to use is 3) – This value is used internally, retry failure counter.

The AudionavDownloaderUtility returns 0 for failure and 1 for Success.

If you need to check what firmware is installed then run the following to retrieve firmware version number

```
AudionavDownloaderUtility -p AUDIONAV -v
```

Change History

Engineering Manual	Date	Version	Details
	29 July 15	1.0	First Release
	12 Aug 15	1.2	Screenshots updated
	01 Sep 15	1.3	API added
	08 Oct 15	1.4	Added amended function for h/v switch on p6
	20 Nov 15	1.5	Added cable tie picture to page 2.
	08 Sep 17	1.6	Update and added Remote Update Instructions
	25 Jan 18	1.7	Added RNIB logo and Externally mounted version

Configuration Utility	Date	Version	Details
	29 Jul 15	2.0	First Release
	08 Sep 17	3.0	Added Win 10 Compatability

Product Firmware	Date	Version	Details
	29/7/15	2.0	Updated so that only vol up / down works as a consumer device.
	10/8/15	4.0	H/V Code table switchover fixed for std table
	25/2/16	5.0	Jack In/Out debounce increased from 400ms to 1.2 sec
	25/3/17	6.0	Improve stability
	18/10/17	7.0	Added 8 digit SN, set LED default brightness to 6, improved recovery process.

Host API Library	Date	Version	Details
	01 Sep 15	1.0	First Release
	08 Sep 17	4.0	Added Win 10 Compatability

Remote Firmware Update	Date	Version	Details
AudioNavDownloaderUtility	08 Sep 17	1.0	New Release, added to Tech Manual