



API Overview	2
Device Communications / Message Format	3
Message Definitions / Error Codes	4
Message Details	5
Change History	22

The content of this communication and / or document, including but not limited to images, specifications, designs, concepts, data and information in any format or medium is confidential and is not to be used for any purpose or disclosed to any third party without the express and written consent of Keymat Technology Ltd. Copyright Keymat Technology Ltd. 2022 .

Storm, Storm Interface, Storm AXS, Storm ATP, Storm IXP , Storm Touchless-CX, AudioNav, AudioNav-EF and NavBar are trademarks of Keymat Technology Ltd. All other trademarks are the property of their respective owners

Storm Interface is a trading name of Keymat Technology Ltd

Storm Interface products include technology protected by international patents and design registration. All rights reserved

API Overview

This section is prepared for application developers who will implement a host application for the NavBar™.

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

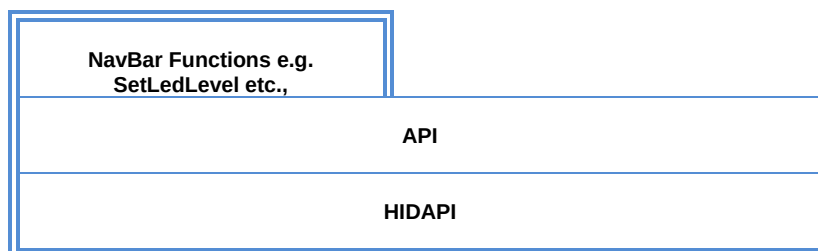
Host API Library - Overview

The Host API Library is a middleware application between the host application and the NavBa. This is available as a download together with the HIDAPI library.

- The API library allows for the host application to invoke the NavBar functions as listed. It 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 NavBar uses USB for communicating with the host. It includes an HID-compliant device. One advantage of using this implementation, which uses only HID interfaces, is that no drivers are required on the host system.

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



With this approach the developer does not need to worry about the communication at low level.

The library can be ported to your specific platform if required.

Currently the library has been tested on Windows and Linux (Ubuntu) platform.

Controlling the NavBar™ from the Host Computer**Device Communications and Message Format**

The NavBar™ & Audio Module 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 a 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

Controlling the NavBar™ from the Host Computer**Message Definitions and Error Codes**

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 NavBar™ – Output the firmware version and all currently selected parameters
02	LED Brightness	Host to NavBar™ – adjust led brightness. (default: 0)
03	Set Jack Led State	Host to NavBar™ – sets the default LED setting for Jack IN/OUT
04	Reserved	Reserved
05	Load New code table	Host to NavBar™ – Load new code table
06	Reserved	Reserved
07	Select Keypad Table	Host to NavBar™ – Select layout table 0 – Default Table 1 – Alternate Table 2 – Customised
08	Reserved	Reserved
09	Write to default	Host to NavBar™ – NavBar™ writes configuration data from ram to flash.
10	Reset to factory default	Host to NavBar™ – Reset device back to factory default
11	RESERVED	Reserved
12	Enable BSL	Host to NavBar™ – Sets the NavBar™ to detect the device loader for firmware loading
13	Reserved	Reserved
14	Reserved	Reserved
15	Retrieve Jack Status	Host to NavBar™ – Retrieves the status of Jack

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

Controlling the Keypad from the Host Computer**List of Messages**

(Structure of Messages from Host to NavBar™ 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	Set Jack Led State	sets the default LED setting for Jack IN/OUT
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 for future use
09	Write to default	NavBar™ 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	Enable BSL	Sets the NavBar™ to detect the device loader for firmware loading
13	Reserved	Reserved for future use
14	Reserved	Reserved for future use
15	Retrieve Jack Status	Retrieves the status of Jack (i.e. JACK IN/OUT)

Structure of Messages from NavBar™ to Host

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

Device Status Request (01)

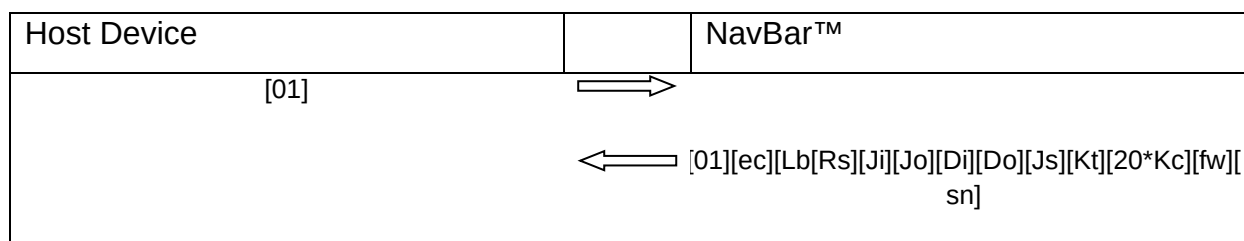
Host sends this message to NavBar™ to request the status of the NavBar™ keypad

NavBar™ Status Response

Secure device 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)
Rs	Reserved	SN	1	Reserved
Ji	LED state Jack IN	SN	1	LED state when Jack IN
Jo	LED state Jack OUT	SN	1	LED state when Jack OUT
Di	LED brightness Jack In	SN	1	LED brightness when Jack IN
Do	LED brightness Jack Out	SN	1	LED brightness when Jack OUT
Js	Jack Status	SN	1	0 – Jack IN, 1 – Jack OUT
Kt	Keypad Table	SN	1	0 – Default Table 1 – Alternate Table 2 – Customised Table
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	YYQQSSSSSSSS. Where YY – year, QQ – Quarter (1-4), SSSSSSSS – unique serial number.

Host sends this message to request information from the NavBar™



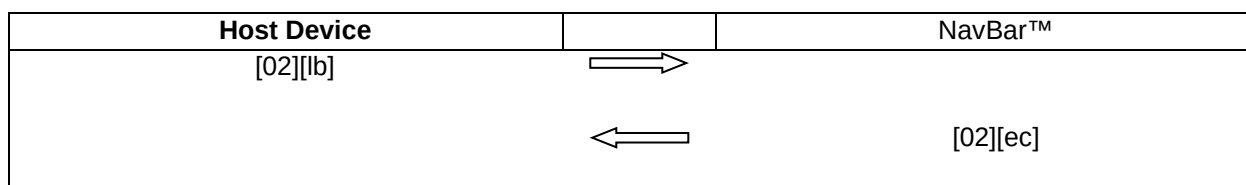
LED Brightness Command (02)

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	



Note: LED brightness of 0 value indicates LEDs are off

LED brightness of 9 value indicates full brightness

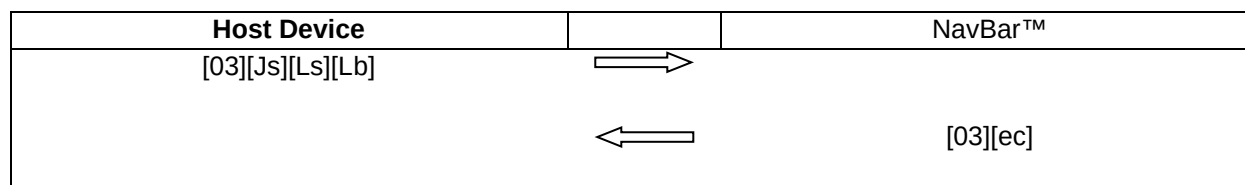
Set Jack LED State (03)

Host sends this message to set Jack IN/OUT led state.

	Data Field	Type	Length	Description
Js	JACK STATE	SN	1	1 – JACK IN, 2 – JACK OUT
Ls	LED STATE	SN	1	Bit 7 6 5 4 3 2 1 0 LED Next Select Down Up Help Back N/A Vol If Bit is set to 1 – LED ON, 0 – LED off
Lb	LED Brightness	SN	1	0 - 9

LED Brightness Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	





Reserved (04)

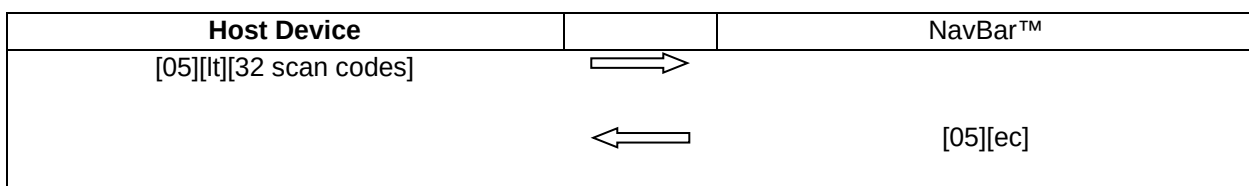
Load New Key Code Table Command (05)

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	



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)

Select Keypad Table Command (07)

Host sends this message to set keypad type

	Data Field	Type	Length	Description
Kt	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		NavBar™
[07][Kt]	→	
	←	[07][ec]



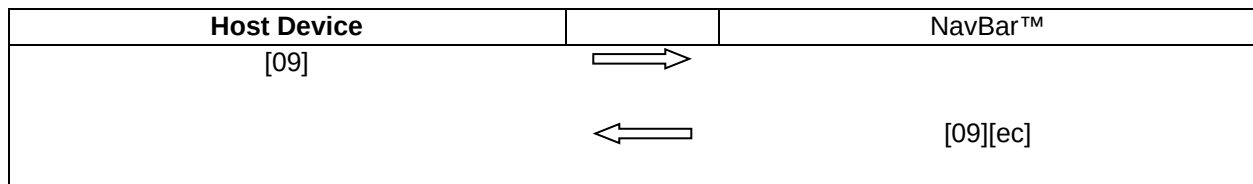
Reserved (08)

Write Config Data To Flash command (09)

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

RAM to FLASH command **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	

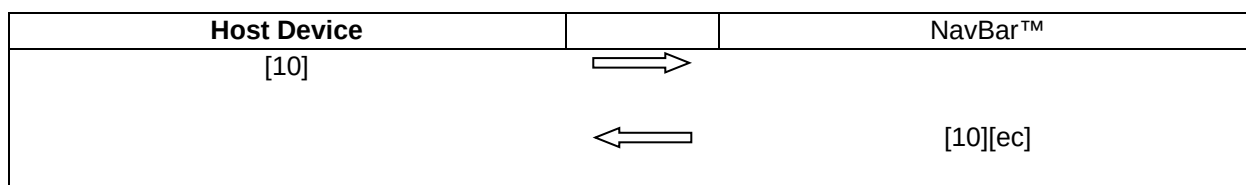


Reset To Factory Default command (10)

Host sends this command to request the NavBar™ 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	





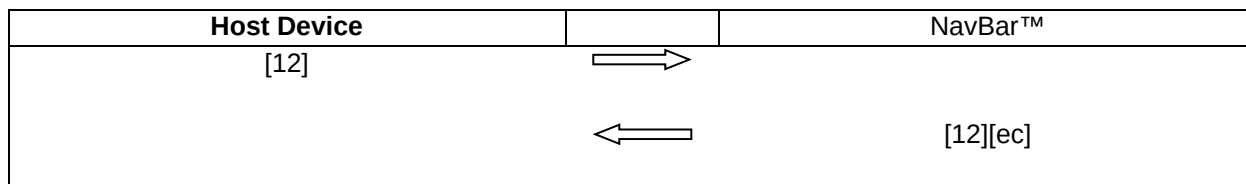
Reserved (11)

Enable BSL Command (12)

Host sends this command to request the NavBar™ to start downloader

Enable BSL command **Response**

	Data Field	Type	Length	Description
ec	Error Code	H	2	





Reserved (13)



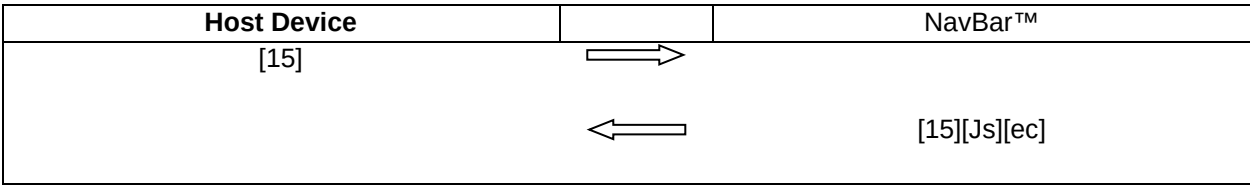
Reserved (14)

Retrieve Jack Status (15)

Host sends this command to request the status of the Jack.

Retrieve Jack Status **Response**

	Data Field	Type	Length	Description
Js	Jack Status	SN	1	0 – Jack Out, 1 – Jack IN
ec	Error Code	H	2	



Key Press Code

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

NavBar™ Report

HID NavBar™ 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 Key_array[0] Key_array[1] Key_array[2] Key_array[3] Key_array[4] Key_array[5]							
Byte2								
Byte3								
Byte4								
Byte5								
Byte6								
Byte7								

For example if user has configured for Default Table. If the user now presses the left key, which is “BACK” and USB code of 70. Then NavBar™ report sent to host would be:

Byte 0 – 0

Byte 1 – 0

Byte 2 – 70

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



Change History

Instructions for API	<u>Date</u>	<u>Version</u>	<u>Details</u>
	15 Aug 24	1.0	First Release (split out from Technical Manual)

Host API Library	<u>Date</u>	<u>Version</u>	<u>Details</u>
	7 Nov 16	1.0	Updated
	08 Sep 17	2.0	Win 10 fix