Contents

Product Overview

This self-contained device is ready to use. It can be easily attached to the rear surface of most Storm keypads to provide connectivity and communication with USB compatible host systems. Factory configured for standard numeric data entry, this versatile device can also be user programmed to output any supported USB code; making the 450 Series encoder the ideal keypad interface for most applications.

Features

Generic keyboard (HID) device – no additional drivers needed
Factory configured to encode telephone or calculator format numeric keypads
Output code table can be customised using Storm's USB Configuration Utility
Host PC can use the supplied API to control the encoder functions in an application.
Integrated power supply for keypad illumination
450i version provides additional brightness control for keypad illumination
450i version features a piezo sounder for optional key press confirmation or application driven status signal
Simple connection via a USB Mini-B socket
Compact, self contained form factor
Compatible with most Storm 4, 12 and 16 key format keypads
(including Storm 700, 720,1000, 2000, 3000, GFX and PLX product series)
Separate ribbon cable available if you do not wish to plug the encoder directly onto the rear of the keypad.

Product Range and Accessories

Part Number	Description
4500-10	450i Encoder with Buzzer and Illumination Control
4500-00	450 Encoder
4500-01	USB Cable 1 metre - type A to angled mini B
4500-CAB07	Ribbon Cable 0.3 metre for remote connection to 4 way keypad
4500-CAB10	Ribbon Cable 0.3 metre for remote connection to 12/16 way keypad

Note :
These part numbers are for on line ordering directly from Storm Interface.
When bought through broadline distribution they have an additional suffix to allow for distributor specific labelling/marking requirements e.g.

4500-102	450i Encoder with Buzzer

Downloads

4500-SW01	USB Configuration Utility
450i-LIT-01	Product Brochure
450-xx-08KT	Installation Sheet
450 USB Manual	Engineering Manual (this document)

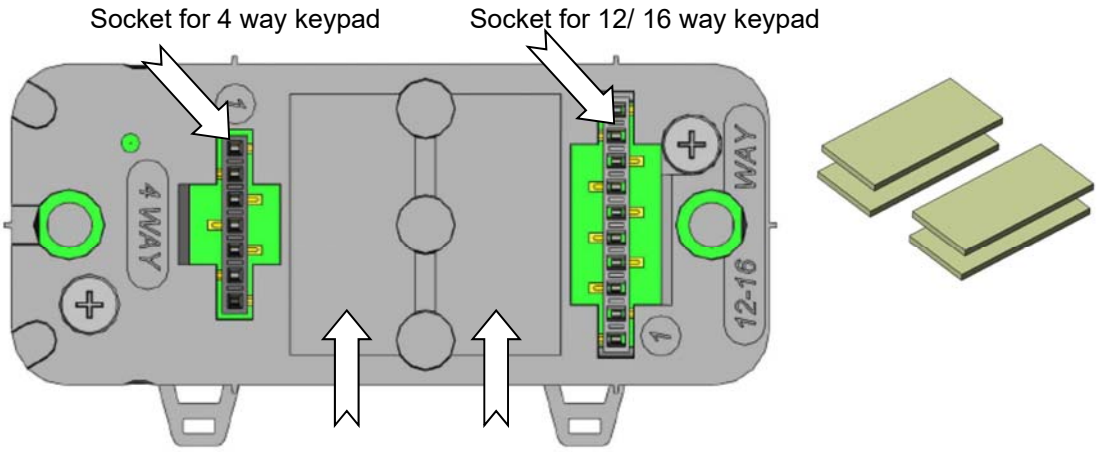
Quickstart Guide

Before starting make sure that you have :

- The encoder
- A compatible Storm keypad.
- A USB mini-B cable between your keypad and the host computer.
- A panel with the correct cutout for your keypad
- A copy of the configuration utility if you want to customise the configuration

Installation

- Ensure your computer is powered up before connecting the encoder.
- Note that there are two different sockets for the keypad connection, depends if a 4 way or 12/16 way keypad is being used. Make sure that the correct socket is used before using the sticky pads to fix the encoder in place.



Sticky pads go here
(one per position, unless using 700 Series underpanel which needs 2 per)

- Push the encoder onto the keypad pins ; make sure the pad sticks down
- Plug in the mini B USB cable on the side of the encoder

One Time Only Initialisation

This initialisation process must be completed the first time you turn it on. The encoder has to recognise the keypad, and you have to select the layout that matches the keypad layout.

a) **PRESS AND HOLD** the bottom right hand key on the keypad – this tells the encoder which keypad is connected

b) Connect the encoder cable to the pc

c) **RELEASE KEY IF** you want function key (4 way) / telephone layout (12/16 way) code table

or

d) **KEEP THE KEY PRESSED FOR 10 SEC IF** you want cursor (4 way) / calculator (12/16 way) code table

Now check that you are getting the correct characters on screen. If you need to reconfigure the encoder you can change the code table (or reset to a pre-loaded code table) with the USB Configuration Utility from www.storm-interface.com

F.A.Q's

Does this encoder need a special driver ?	No – it works with the standard USB keyboard driver
Does the utility work on any pc ?	At present it does not run on Linux or Mac os The utility requires Windows XP or later
What's the USB connection ?	Mini-B socket
Do I need to use the sticky pads ?	These are included to retain the encoder in service
What custom USB codes can I assign ?	See the code tables on page 11
What do I do if I have wrongly initialised the product ?	Download & use the config utility to reset the defaults
Why is the socket longer than the pinstrip on my keypad?	The end pins power the 720 illuminated keypads.
Can I control this from a host application ?	Yes – the commands are listed in the API reference

Ratings & Performance

Operational temperature	-20ºC to +60ºC
Storage temperature	-20ºC to +70ºC
Humidity	10% to 90% non-condensing
Vibration and shock	ETSI 300 019 5M3
Operating voltage	5V +/- 5% (USB)
Operating current	20mA (excluding keypad illumination current)
Safety	EU Low Voltage Directive
EMC:	Emissions and Immunity: FCC part 15 class A
	EN55022, EN55024
	ESD: Up to +/- 15kV air discharge, +/- 7.5kV contact discharge
EU RoHS	
WEEE Directive compliant	

Compatible Products

	4 Key	12 Key	16 Key	Note
700 Series	☑	☑	☑	Use additional sticky pads for underpanel fixing of 700 Series
720 Series	☑	☑	☑	720 and 720 illuminated keypads are supported
1000 Series	☑	☑	☑	
PLX Series	☑	☑	☑	
2000 Series	☑	☑	☑	
GFX	☑	☑	☑	
3000 Series	☑	☑	☑	
SF6000			☑	Use remote connection cable.(NB Illumination circuit is separate)
3000 Illuminator				Illumination not supported on this model
GFX Illuminator				Illumination not supported on this model
	Use the 7 way socket for 4 key pad	Use the 10 way socket for 12/16 key pad		

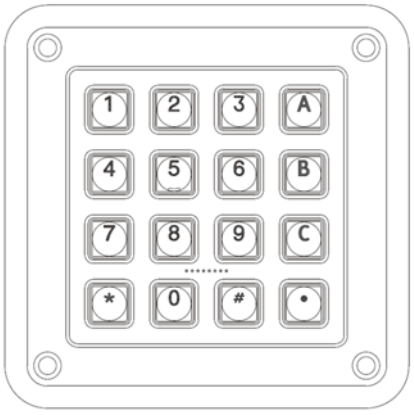
Keypad Layouts

Keypad Layouts

Default Code Table (US English)

4-Way Function

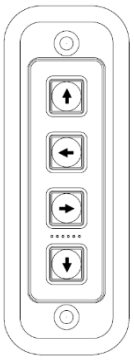
12-Way Telephone



16-Way Telephone

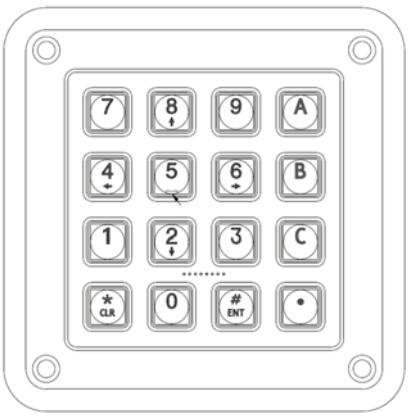
Keypad Layouts

Alternate Code Table (US English)



4-Way Cursor

12-Way Calculator



16-Way Calculator

Default Code Tables (remember host set to UK English gives £ instead of #)

Row	Column	4 way codes Function (hex)	12 way code Telephone(hex)	16 way code Telephone(hex)
A	1	F1 (3A)	1 (1E)	1 (1E)
B	1	F2 (3B)	4 (21)	4 (21)
C	1	F3 (3C)	7 (24)	7 (24)
D	1	F4 (3D)	* (E1, 25)	* (E1, 25)
A	2	-	2 (1F)	2 (1F)
B	2	-	5 (22)	5 (22)
C	2	-	8 (25)	8 (25)
D	2	-	0 (27)	0 (27)
A	3	-	3 (20)	3 (20)
B	3	-	6 (23)	6 (23)
C	3	-	9 (26)	9 (26)
D	3	-	# (E1, 20)	# (E1, 20)
A	4	-	-	A (04)
B	4	-	-	B (05)
C	4	-	-	C (06)
D	4	-	-	. (37)

Alternate Code Table (to get the arrow keys on a 12/16 way keypad then switch NumLock off)

Row	Column	4 way code Cursor (hex)	12 way code Calculator (hex)	16 way code Calculator (hex)	Output for 12/16 way with NumLock off
A	1	↑ (52)	7 (5F)	7 (5F)	HOME
B	1	← (50)	4 (5C)	4 (5C)	←
C	1	→ (4F)	1 (59)	1 (59)	END
D	1	↓ (51)	* (E1, 25)	* (E1, 25)	*
A	2	-	8 (60)	8 (60)	↑
B	2	-	5 (5D)	5 (5D)	
C	2	-	2 (5A)	2 (5A)	↓
D	2	-	0 (62)	0 (62)	
A	3	-	9 (61)	9 (61)	PgUp
B	3	-	6 (5E)	6 (5E)	→
C	3	-	3 (5B)	3 (5B)	PgDn
D	3	-	# (E1, 20)	# (E1, 20)	#
A	4	-	-	A (04)	A
B	4	-	-	B (05)	B
C	4	-	-	C (06)	C
D	4	-	-	. (37)	.

Configuration Utility

To customise the output codes just download and install the Configuration Utility from www.storm-interface.com
This lets you do the following :-

Scan the encoder in order to	Confirm the encoder is connected Show which version of firmware is installed Show which keypad is set (4, 12 or 16 key) Show which code table is selected (default, alternate or customised)
And also	Change the keypad setting Change the selected code table Change the buzzer volume (450i only) Change the brightness on illuminated keypads (450i only) Self test the encoder
For re-legendable keypads	Customise the code table by assigning a USB code to each key Add a modifier in front of each USB code Save this configuration Export or Import configuration files
For maintenance purposes	Update the encoder firmware if a new version is released Restore all settings to original factory defaults.

API

To allow a host application to control the USB encoder the available commands are listed in the API Documentation. Free download from www.storm-interface.com

Configuration Utility User Guide

Download from www.storm-interface.com and install on a Windows PC with XP or later

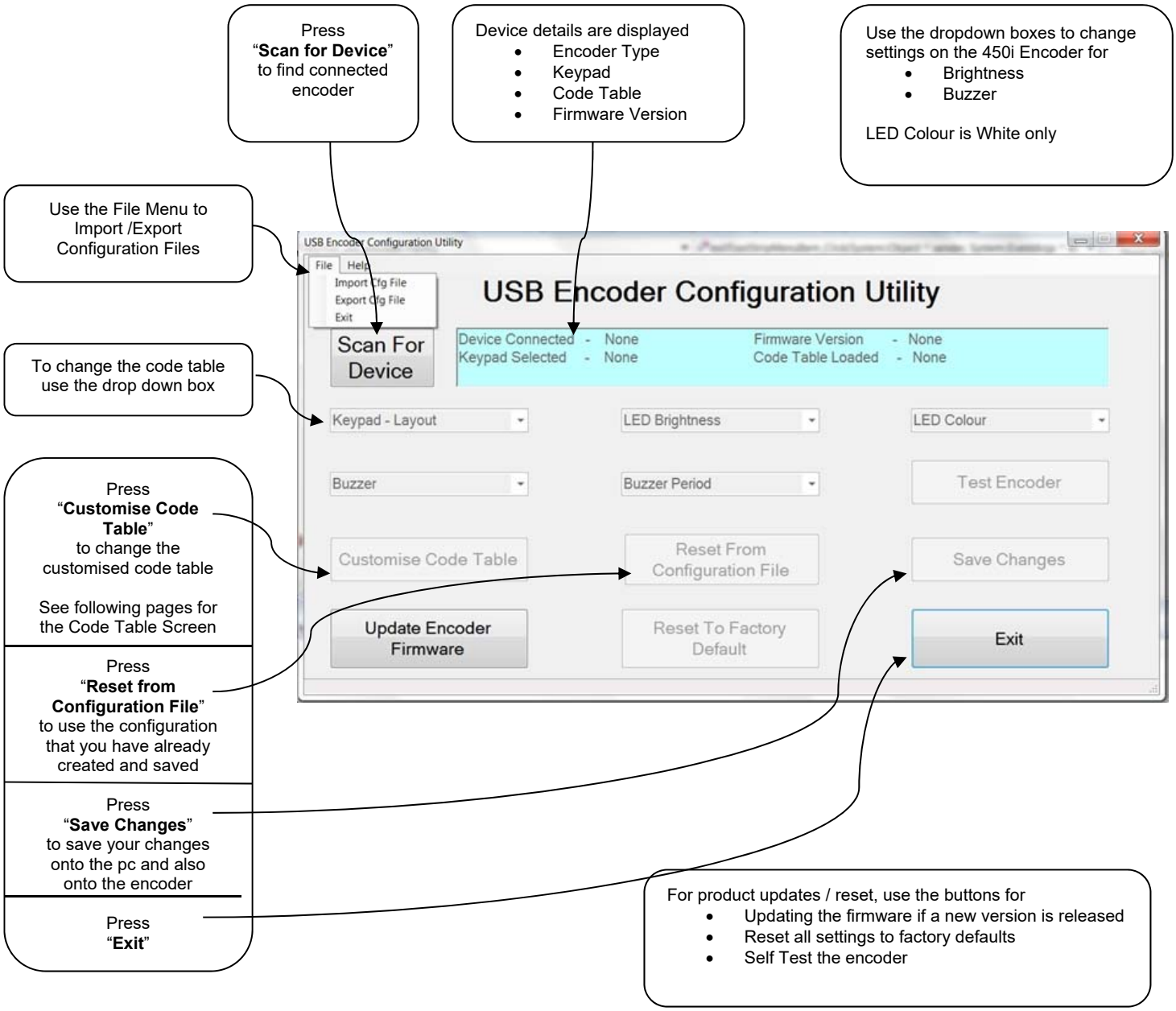
Run the application.

Plug in the encoder + keypad.

Scan the encoder. The configuration will be displayed as below on the home screen.

If you have a standard layout keypad then the output from the default code table will correspond to the keypad
If you have a keypad designed to allow customisation of keytop graphics then you need to assign a code to each key.

The configuration file is saved to the pc and to the encoder when the **Save Changes** button is pressed.



Configuration Utility User Guide

Customising the Code Table

The utility displays a screen that shows for each key

- Which USB code is assigned
- Which modifier (if any) is applied to the USB code.

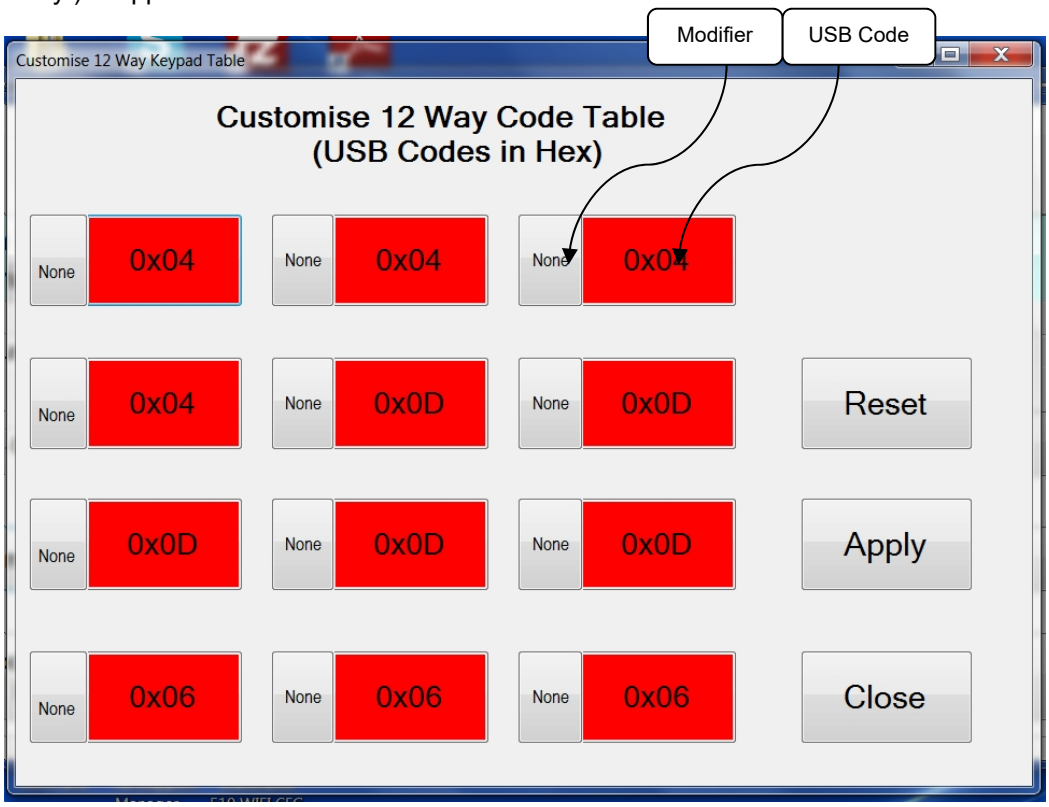
Click on each position and select a USB code from the drop down list.

Add a modifier for each position if required.

Press "**Apply**" to reserve your changes. *This does not save the changes at this stage.*

Press "**Close**" to return to the home screen

"**Reset**" reloads the default code table



The full list of USB Codes is shown on the following pages.
USB Codes that have been checked in Word are shown in the relevant column, for example :

			Un-shifted	Shifted
Code	0x04	gives	a	A

Where the same USB code gives a different character dependent on the host language setting then this is shown in the relevant language column.

The actual function of the USB code is determined by the application; not all codes have a function in every application.

Configuration Utility User Guide

Updating the Firmware

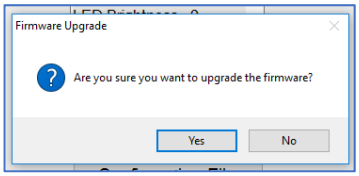
When you update the firmware the utility keeps a copy of your configuration (including any customised codes), and reloads it back on the encoder as part of the firmware update process

Download the new firmware from www.storm-interface.com,

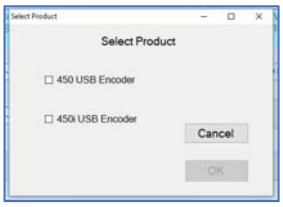
Connect the encoder.

Press **Scan for Device** to find the connected encoder

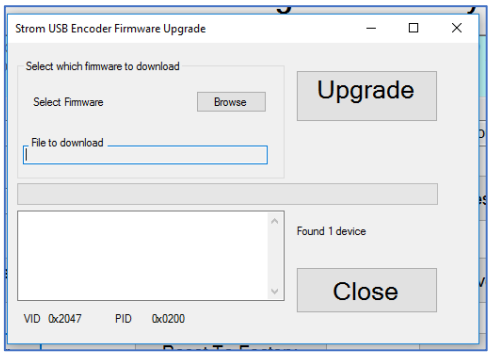
Press **Update Encoder Firmware** and press **Yes**



Select encoder type and press **OK**

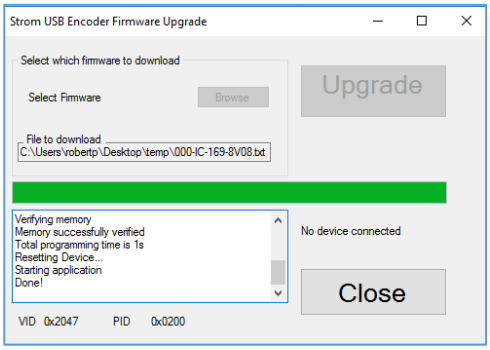


Browse to locate the firmware file and press **Upgrade**



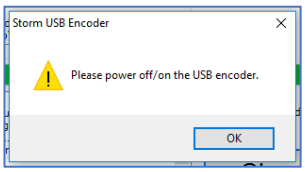
The progress bar shows in green.

When progress is complete press **Close**

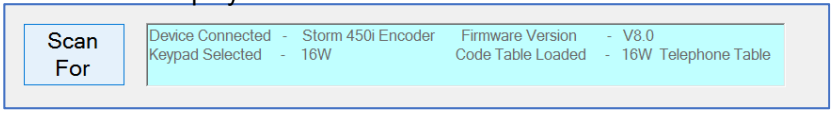


Unplug the cable

Re-connect the cable and press **OK**



Press **Scan For** and the new version of firmware will be displayed



Full Code Table Reference

450 Series USB Encoder with Firmware Revision 8v04
Using Generic HID Keyboard Driver

When customising the code table on the encoder you can place a modifier in front of the USB Code

e.g. E1 , 34 will give you @

				Any Language differences (using Word)								
				English U.K. (if different to U.S.)		English U.S.			French	German	Spanish
USB Usage ID (Dec)	**USB Usage ID (Hex)**	**Usage Name**	Note	Un-shifted	Shifted	Un-shifted		Shifted	Num lock			
00	00	Reserved (no event indicated)	9									
01	01	Keyboard Error Roll Over	9									
02	02	Keyboard POST Fail	9									
03	03	Keyboard Error Undefined	9									
04	04	Keyboard a and A	4			a		A				
05	05	Keyboard b and B				b		B				
06	06	Keyboard c and C	4			c		C				
07	07	Keyboard d and D				d		D				
08	08	Keyboard e and E				e		E				
09	09	Keyboard f and F				f		F				
10	0A	Keyboard g and G				g		G				
11	0B	Keyboard h and H				h		H				
12	0C	Keyboard i and I				i		I				
13	0D	Keyboard j and J				j		J				
14	0E	Keyboard k and K				k		K				
15	0F	Keyboard l and L				l		L				
16	10	Keyboard m and M	4			m		M				
17	11	Keyboard n and N				n		N				
18	12	Keyboard o and O	4			o		O				
19	13	Keyboard p and P	4			p		P				
20	14	Keyboard q and Q	4			q		Q				
21	15	Keyboard r and R				r		R				
22	16	Keyboard s and S	4			s		S				
23	17	Keyboard t and T				t		T				
24	18	Keyboard u and U				u		U				
25	19	Keyboard v and V				v		V				
26	1A	Keyboard w and W	4			w		W				
27	1B	Keyboard x and X	4			x		X				
28	1C	Keyboard y and Y	4			y		Y				
29	1D	Keyboard z and Z	4			z		Z				
30	1E	Keyboard 1 and !	4			1		!				
31	1F	Keyboard 2 and @	4	2	"	2		@				
32	20	Keyboard 3 and #	4	3	£	3		#				
33	21	Keyboard 4 and $	4			4		$				
34	22	Keyboard 5 and %	4			5		%				
35	23	Keyboard 6 and ^	4			6		^				
36	24	Keyboard 7 and &	4			7		&				
37	25	Keyboard 8 and *	4			8		*				
38	26	Keyboard 9 and (	4			9		(				
39	27	Keyboard 0 and)				0		)				

450 Series USB Encoder with Firmware Revision 8v04
Using Generic HID Keyboard Driver

When customising the code table on the encoder you can place a modifier in front of the USB Code

e.g. E1 , 34 will give you @

				Any Language differences (using Word)								
				English U.K. (if different to U.S.)		English U.S.			French	German	Spanish
USB Usage ID (Dec)	**USB Usage ID (Hex)**	**Usage Name**	Note	Un-shifted	Shifted	Un-shifted		Shifted	Num lock			
40	28	Keyboard Return (ENTER)	5									
41	29	Keyboard ESCAPE										
42	2A	Keyboard DELETE (Backspace)	13									
43	2B	Keyboard Tab										
44	2C	Keyboard Spacebar										
45	2D	Keyboard - and (underscore)4	4			-		_				
46	2E	Keyboard = and +	4			=		+				
47	2F	Keyboard [and {	4			[		{				
48	30	Keyboard] and }	4			]		}				
49	31	Keyboard \ and |				\		|				
50	32	Keyboard Non-US # and ~	2	#	~	\		|				
51	33	Keyboard ; and :	4			;		:				
52	34	Keyboard ‘ and “	4	‘	@	‘		“				
53	35	Keyboard Grave Accent and Tilde	4			`		~				
54	36	Keyboard, and <	4			,		<				
55	37	Keyboard . and >	4			.		>				
56	38	Keyboard / and ?	4			/		?				
57	39	Keyboard Caps Lock11	11									
58	3A	Keyboard F1				F1						
59	3B	Keyboard F2				F2						
60	3C	Keyboard F3				F3						
61	3D	Keyboard F4				F4						
62	3E	Keyboard F5				F5						
63	3F	Keyboard F6				F6						
64	40	Keyboard F7				F7						
65	41	Keyboard F8				F8						
66	42	Keyboard F9				F9						
67	43	Keyboard F10				F10						
68	44	Keyboard F11				F11						
69	45	Keyboard F12				F12						
70	46	Keyboard PrintScreen	1									
71	47	Keyboard Scroll Lock	11									
72	48	Keyboard Pause	1									
73	49	Keyboard Insert	1									
74	4A	Keyboard Home	1			Home		Select line of text				
75	4B	Keyboard PageUp	1			PgUp		Select text above				
76	4C	Keyboard Delete Forward	1,14			Delete		Select text forward				
77	4D	Keyboard End	1			End		Select to end				
78	4E	Keyboard PageDown	1			PgDn		Select to page down				

450 Series USB Encoder with Firmware Revision 8v04
Using Generic HID Keyboard Driver

When customising the code table on the encoder you can place a modifier in front of the USB Code

e.g. E1 , 34 will give you @

				Any Language differences (using Word)								
				English U.K. (if different to U.S.)		English U.S.				French	German	Spanish
USB Usage ID (Dec)	**USB Usage ID (Hex)**	**Usage Name**	Note	Un-shifted	Shifted	Un-shifted		Shifted	Num lock			
79	4F	Keyboard RightArrow	1			Goes right		Select to right				
80	50	Keyboard LeftArrow	1			Goes left		Select to left				
81	51	Keyboard DownArrow	1			Goes down		Select line down				
82	52	Keyboard UpArrow	1			Goes up		Select line up				
83	53	Keypad Num Lock and Clear	11			Toggles Numlock						
84	54	Keypad /	1			/						
85	55	Keypad *				*						
86	56	Keypad -				-						
87	57	Keypad +				+						
88	58	Keypad ENTER				Enter						
89	59	Keypad 1 and End				End			1			
90	5A	Keypad 2 and Down Arrow				Down arrow			2			
91	5B	Keypad 3 and PageDn				Page down			3			
92	5C	Keypad 4 and Left Arrow				Left arrow			4			
93	5D	Keypad 5							5			
94	5E	Keypad 6 and Right Arrow				Right arrow			6			
95	5F	Keypad 7 and Home				Home			7			
96	60	Keypad 8 and Up Arrow				Up arrow			8			
97	61	Keypad 9 and PageUp				Page up			9			
98	62	Keypad 0 and Insert							0			
99	63	Keypad . and Delete				.			.			
100	64	Keyboard Non-US \ and |	3,6			\		|				
101	65	Keyboard Application	12									
102	66	Keyboard Power	9									
103	67	Keypad =				= on Mac O/S only						
104	68	Keyboard F13										
105	69	Keyboard F14										
106	6A	Keyboard F15										
107	6B	Keyboard F16										
108	6C	Keyboard F17										
109	6D	Keyboard F18										
110	6E	Keyboard F19										
111	6F	Keyboard F20										
112	70	Keyboard F21										
113	71	Keyboard F22										
114	72	Keyboard F23										

450 Series USB Encoder with Firmware Revision 8v04
Using Generic HID Keyboard Driver

When customising the code table on the encoder you can place a modifier in front of the USB Code

e.g. E1 , 34 will give you @

USB Usage ID (Dec)	USB Usage ID (Hex)	Usage Name	Note	English U.K. (if different to U.S.) Un-shifted	English U.K. Shifted	English U.S. Un-shifted	English U.S. Shifted	English U.S. Num lock	French	German	Spanish
115	73	Keyboard F24									
116	74	Keyboard Execute									
117	75	Keyboard Help									
118	76	Keyboard Menu									
119	77	Keyboard Select									
120	78	Keyboard Stop									
121	79	Keyboard Again									
122	7A	Keyboard Undo									
123	7B	Keyboard Cut									
124	7C	Keyboard Copy									
125	7D	Keyboard Paste									
126	7E	Keyboard Find									
127	7F	Keyboard Mute									
128	80	Keyboard Volume Up									
129	81	Keyboard Volume Down									
130	82	Keyboard Locking Caps Lock	12								
131	83	Keyboard Locking Num Lock	12								
132	84	Keyboard Locking Scroll Lock	12								
133	85	Keypad Comma	27								
134	86	Keypad Equal Sign	29								
135	87	Keyboard International115									
136	88	Keyboard International216									
137	89	Keyboard International317									
138	8A	Keyboard International418									
139	8B	Keyboard International519									
140	8C	Keyboard International620									
141	8D	Keyboard International721									
142	8E	Keyboard International822									
143	8F	Keyboard International922									
144	90	Keyboard LANG125									
145	91	Keyboard LANG226									
146	92	Keyboard LANG330									
147	93	Keyboard LANG431									

Any Language differences (using Word)

450 Series USB Encoder with Firmware Revision 8v04
Using Generic HID Keyboard Driver

When customising the code table on the encoder you can place a modifier in front of the USB Code

e.g. E1 , 34 will give you @

				Any Language differences (using Word)								
				English U.K. (if different to U.S.)		English U.S.				French	German	Spanish
USB Usage ID (Dec)	**USB Usage ID (Hex)**	**Usage Name**	Note	Un-shifted	Shifted	Un-shifted		Shifted	Num lock			
148	94	Keyboard LANG532										
149	95	Keyboard LANG68										
150	96	Keyboard LANG78										
151	97	Keyboard LANG88										
152	98	Keyboard LANG98										
153	99	Keyboard Alternate Erase7										
154	9A	Keyboard SysReq/Attention1										
155	9B	Keyboard Cancel										
156	9C	Keyboard Clear										
157	9D	Keyboard Prior										
158	9E	Keyboard Return										
159	9F	Keyboard Separator										
160	A0	Keyboard Out										
161	A1	Keyboard Oper										
162	A2	Keyboard Clear/Again										
163	A3	Keyboard CrSel/Props										
164	A4	Keyboard ExSel										
224	E0	Keyboard LeftControl										
225	E1	Keyboard LeftShift										
226	E2	Keyboard LeftAlt										
227	E3	Keyboard Left GUI	10,23									
228	E4	Keyboard RightControl										
229	E5	Keyboard RightShift										
230	E6	Keyboard RightAlt										
231	E7	Keyboard Right GUI	10.24									

Notes on the Code Tables 1-15, 20-34

1 Usage of keys is not modified by the state of the Control, Alt, Shift or Num Lock keys. That is, a key does not send extra codes to compensate for the state of any Control, Alt, Shift or Num Lock keys.

2 Typical language mappings: US: \| Belg: ƒÊ` ’ FrCa: <}> Dan: f* Dutch: <> Fren:*ƒÊ Ger: # f Ital: u ˜ LatAm: }`] Nor:,* Span: }C Swed: ,* Swiss: $ ’ UK: #~.

3 Typical language mappings: Belg:<\> FrCa: á ‹ â Dan:<\> Dutch:]|[Fren:<> Ger:<|> Ital:<> LatAm:<> Nor:<> Span:<> Swed:<|> Swiss:<\> UK:\| Brazil: \|.

4 Typically remapped for other languages in the host system.

5 Keyboard Enter and Keypad Enter generate different Usage codes.

6 Typically near the Left-Shift key in AT-102 implementations.

7 Example, Erase-Eaze. key.

8 Reserved for language-specific functions, such as Front End Processors and Input Method Editors.

9 Reserved for typical keyboard status or keyboard errors. Sent as a member of the keyboard array. Not a physical key.

10 Windows key for Windows 95, and gCompose. h

11 Implemented as a non-locking key; sent as member of an array.

12 Implemented as a locking key; sent as a toggle button. Available for legacy support; however, most systems should use the non-locking version of this key.

13 Backs up the cursor one position, deleting a character as it goes.

14 Deletes one character without changing position.

15-20 See additional foot notes in the USB spec

21 Toggle double-byte/single-byte mode

22 Undefined, available for other front end language processors

23 Windowing environment key, examples are Microsoft left win key, mac left apple key, sun left meta key

24 Windowing environment key, example are microdoft wight win key, macintosh right apple key, sun right meta key

Product Dimensions

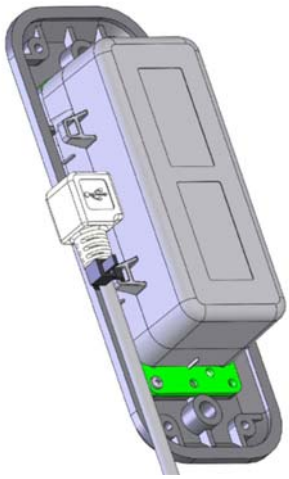
Overall dims	77mm x 39mm x 25mm, 30 grams
Packed dims	124mm x 52mm x 40mm, 50 grams
Included parts	Qty 4 sticky pads, Installation sheet

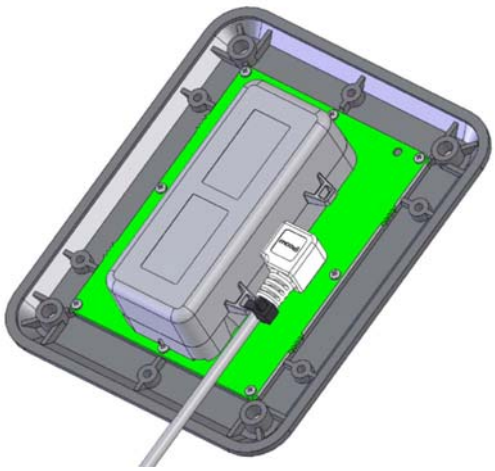
Cables

Depending on your installation you may require either a straight or an angled cable, with latching mini B USB connector. If you use an angled cable then you will be able to secure the cable to the encoder as below.
Note : Angled Mini B cables are available with 2 possible connectors – see picture below.

Encoder on 4 way keypad

Encoder on 12 way keypad
(order pn 4500-01 from Storm)



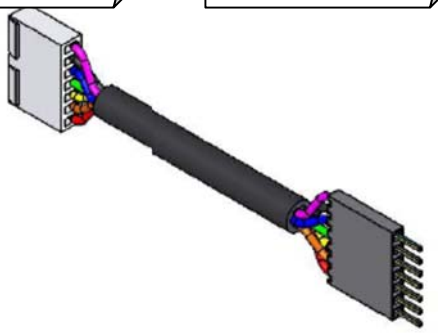


If you wish to install the encoder remote from the keypad then you will need an interconnection cable
Cable details as shown below (or order 7 way cable PN 4500-CAB07 ,10 way cable PN 4500-CAB10)

Encoder 4 W		4 w keypad	Illuminated version
PIN	to	PIN	PIN
1			1
2		1	2
3		2	3
4		3	4
5		4	5
6		5	6
7			7

Molex KK 0.1”
Pitch Socket
2695 Series

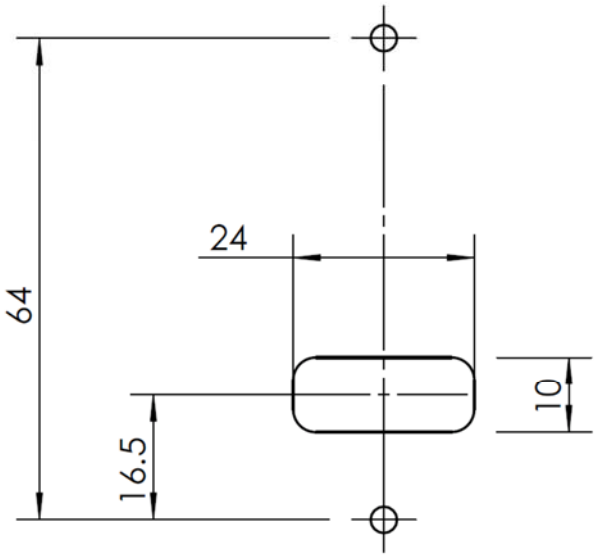
Molex KK 0.1”
Pitch PIN
Headers



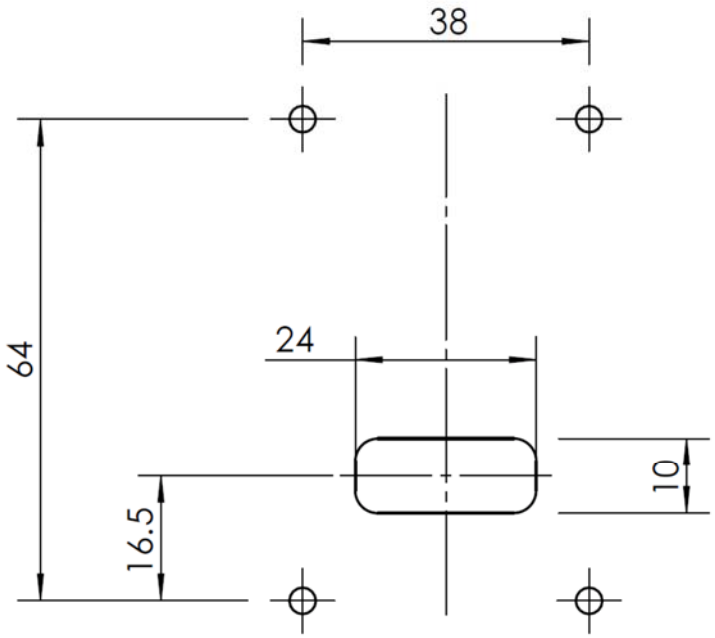
	Encoder 12/16		12/16 keypad	Illuminated version	SF6000 Series Matrix Keypad
	PIN	to	PIN	PIN	PIN
	1			1	1
R1	2		1	2	2
R2	3		2	3	3
C1	4		3	4	4
C2	5		4	5	5
C3	6		5	6	6
C4	7		6	7	7
R4	8		7	8	8
R3	9		8	9	9
	10			10	10

Panel Cutout Drawings

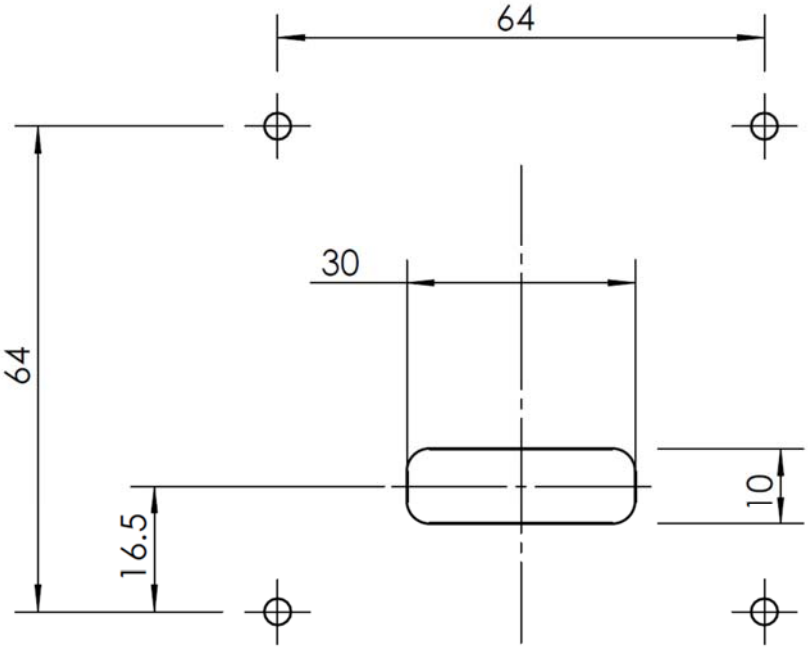
Surface Fixing of Keypads : 700 Series. 720 Series Both products use the same panel cutout detail.



4 WAY PANEL FRONT MOUNTED



12 WAY PANEL FRONT MOUNTED



16 WAY PANEL FRONT MOUNTED

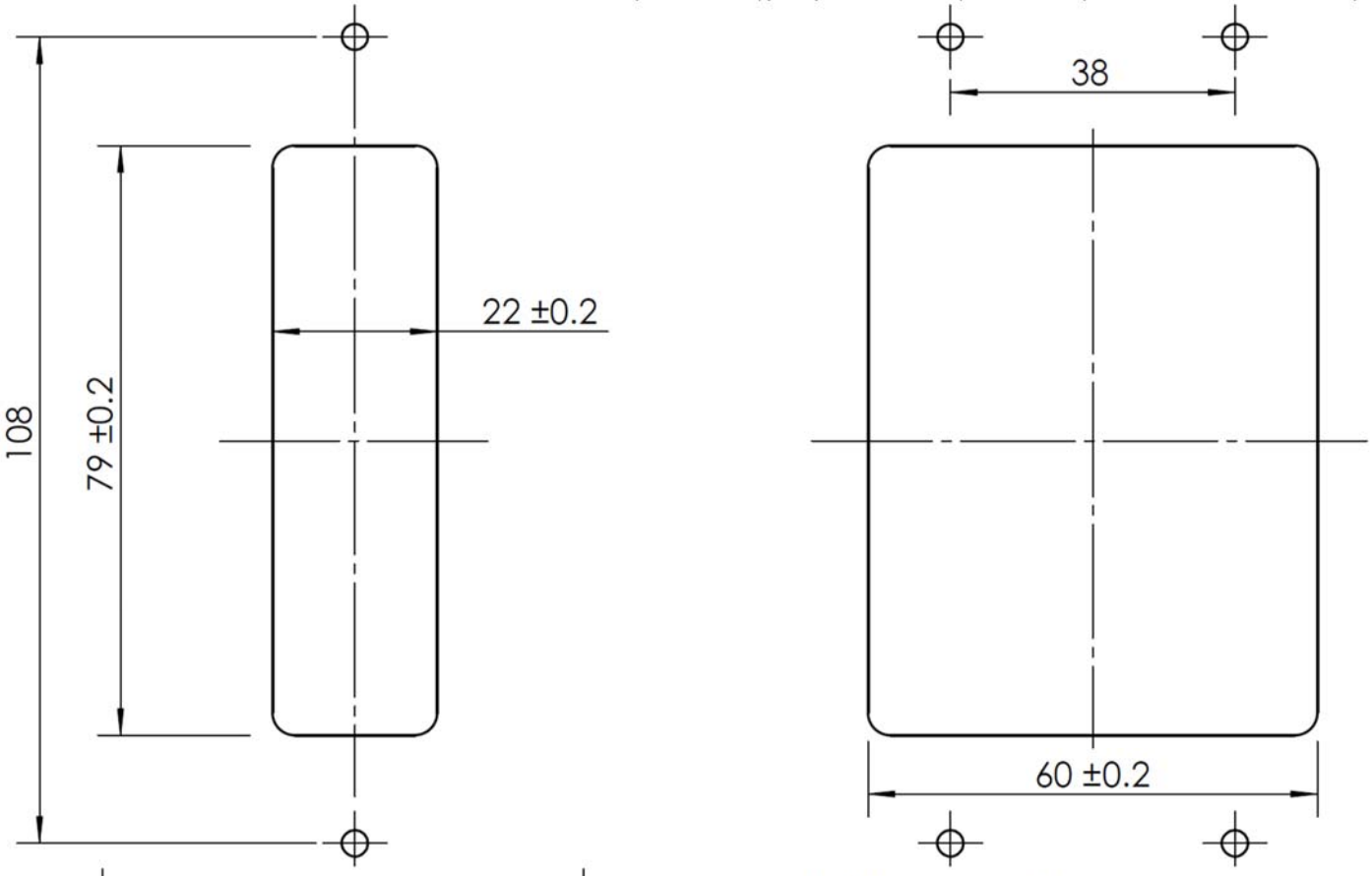
PANEL DETAILS FOR
700 AND 720 SERIES
FRONT FIXING

ALL DIMS IN MM
R3 IN CORNERS

CLEARANCE HOLES
DIAMETER 3.5 FOR STUDS

Panel Cutout Drawings

Underpanel Fixing of Keypads : 700 Series. (4 way uses fix kit 7004CL0, 12/16 way uses fix kit 7012CL0)
720 Series (uses fixing kit pn 7204CL0, 12/16 way uses fix kit 7212CL0)



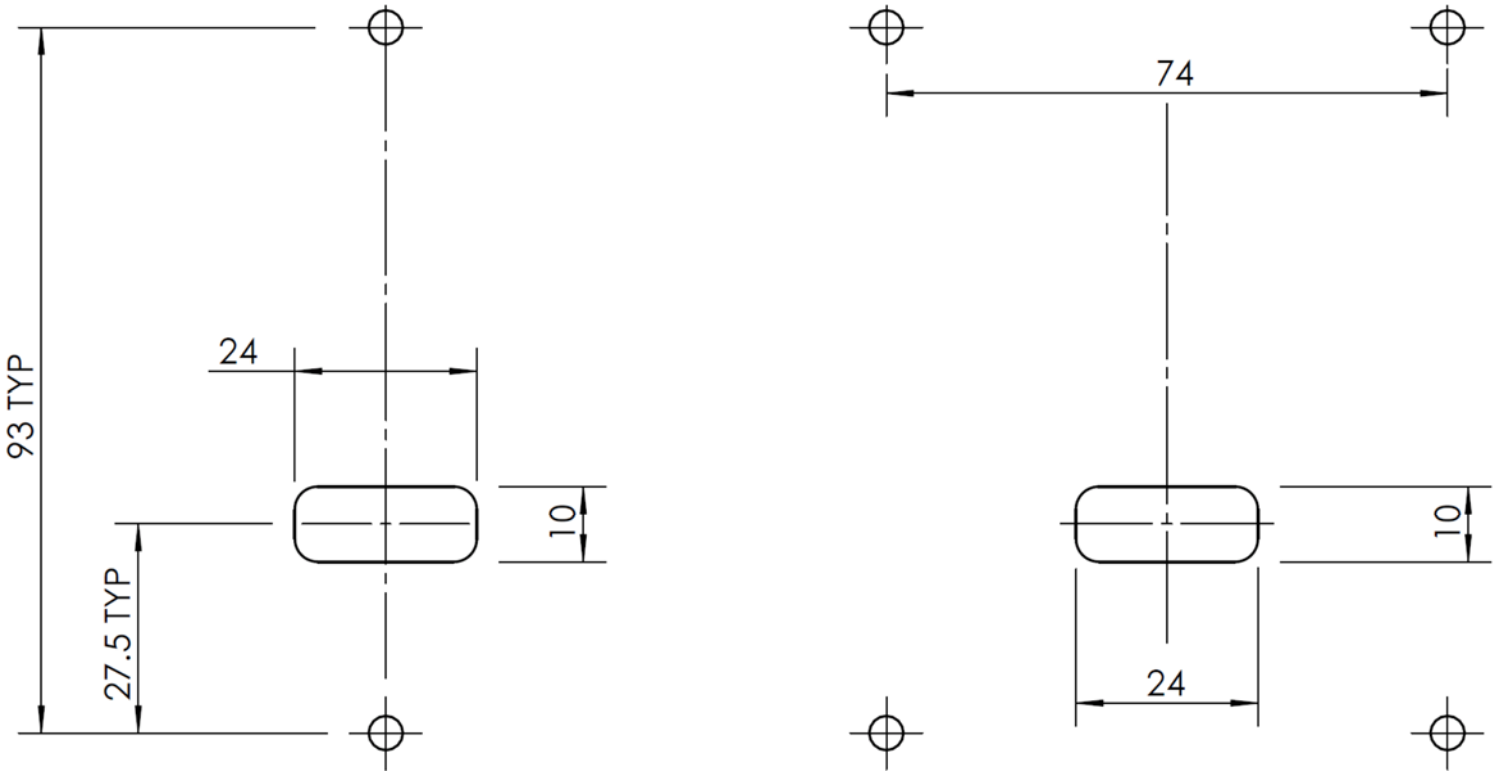
PANEL DETAILS FOR
700 AND 720 SERIES
UNDERPANEL FIXING

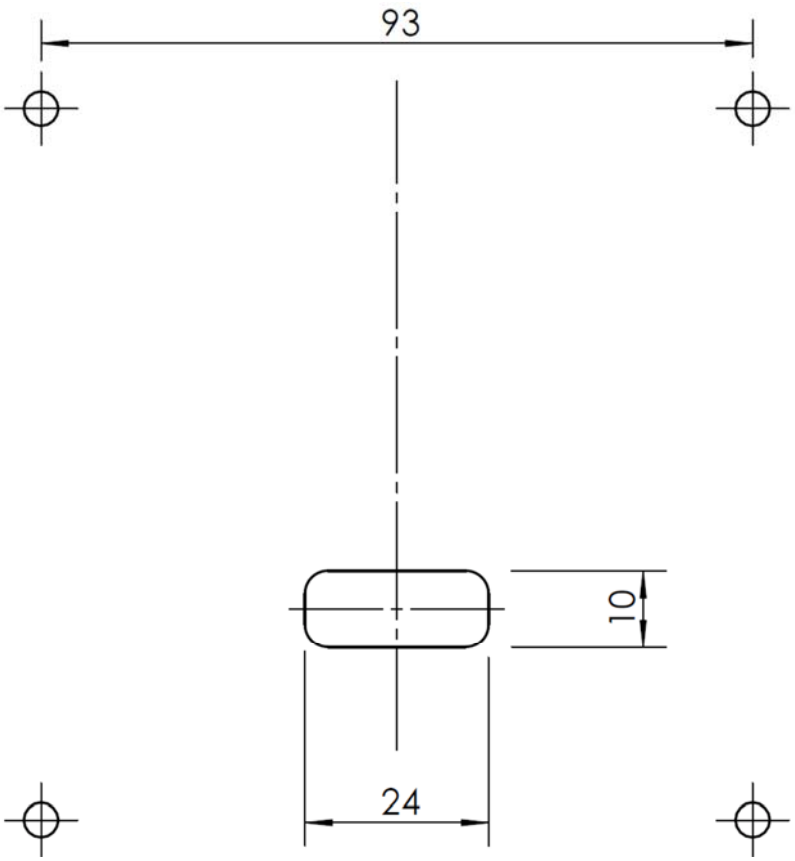
4, 12 AND 16 WAY
KEYPADS
MAKE SURE TO USE THE
CORRECT FIXING CLIPS
FOR PRODUCT

ALL DIMS IN MM
R3 IN CORNERS
M3 STUDS OR SIMILAR

Panel Cutout Drawings

Surface Fixing of Keypads : 1000 Series, 2000 Series, PLX Series





PANEL DETAILS FOR
1000 AND 200 SERIES
FRONT FIXING

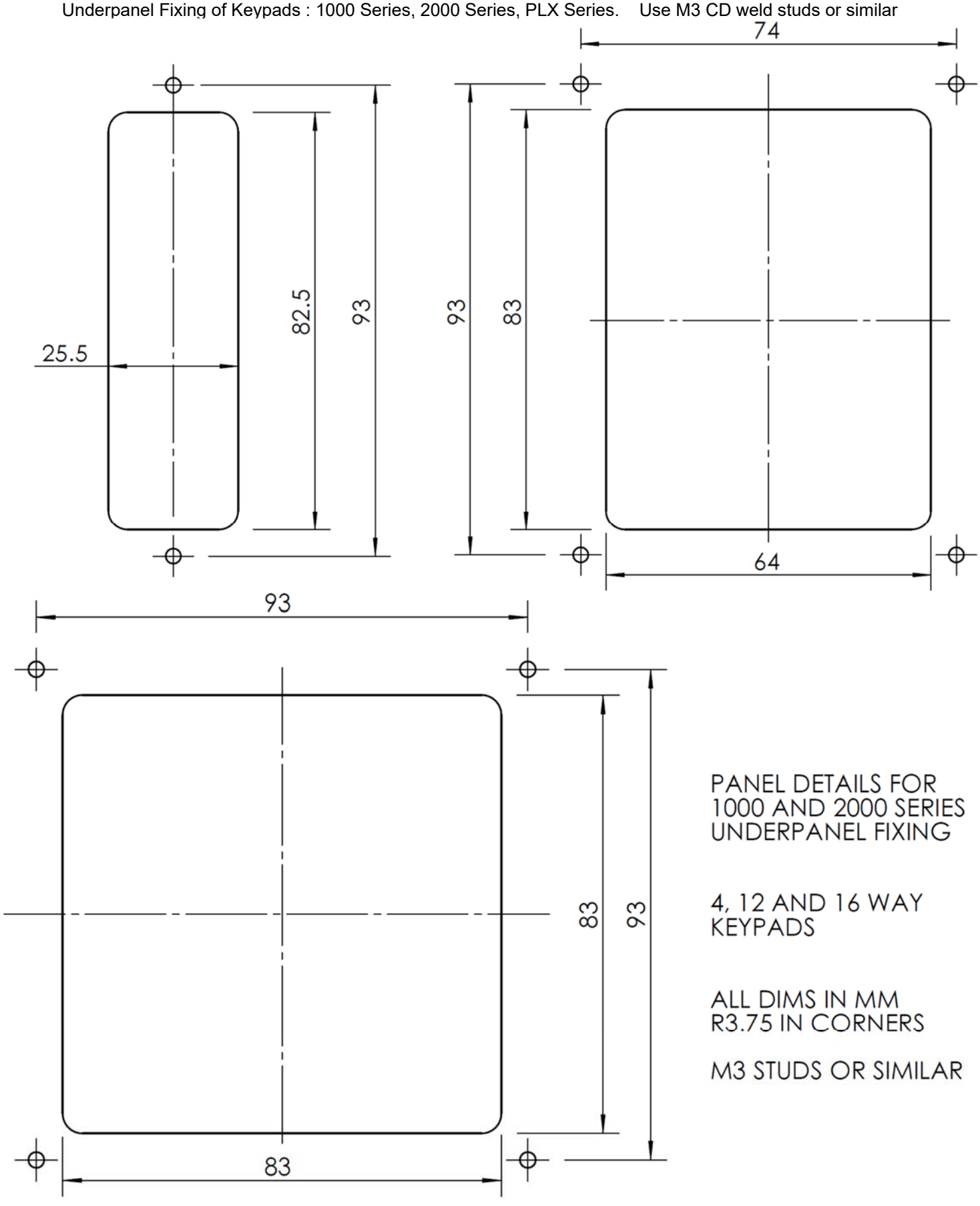
4, 12 AND 16 WAY
KEYPADS

ALL DIMS IN MM
R3 IN CORNERS

CLEARANCE HOLES
DIAMETER 4.5

Panel Cutout Drawings

Underpanel Fixing of Keypads : 1000 Series, 2000 Series, PLX Series. Use M3 CD weld studs or similar



PANEL DETAILS FOR
1000 AND 2000 SERIES
UNDERPANEL FIXING

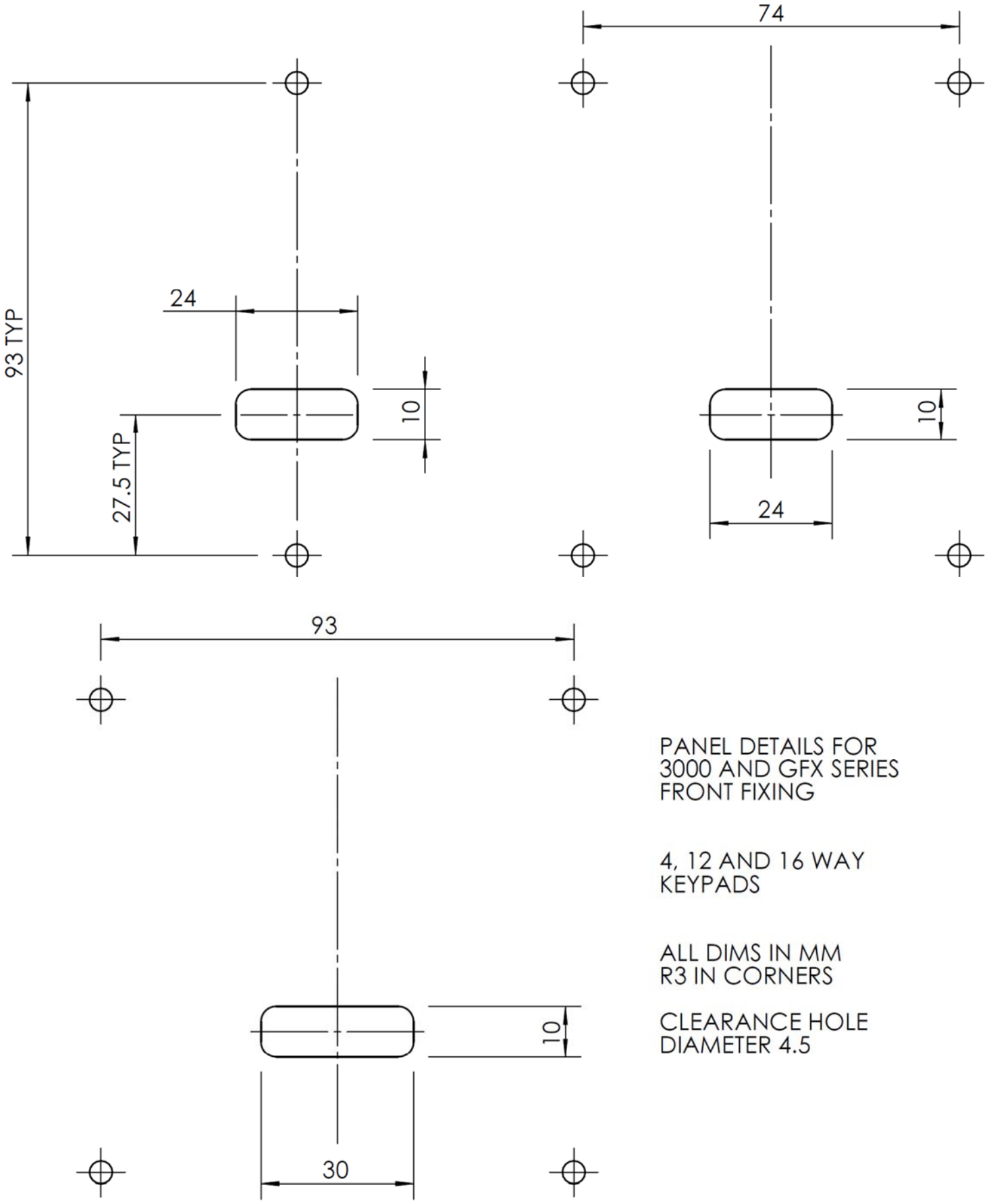
4, 12 AND 16 WAY
KEYPADS

ALL DIMS IN MM
R3.75 IN CORNERS

M3 STUDS OR SIMILAR

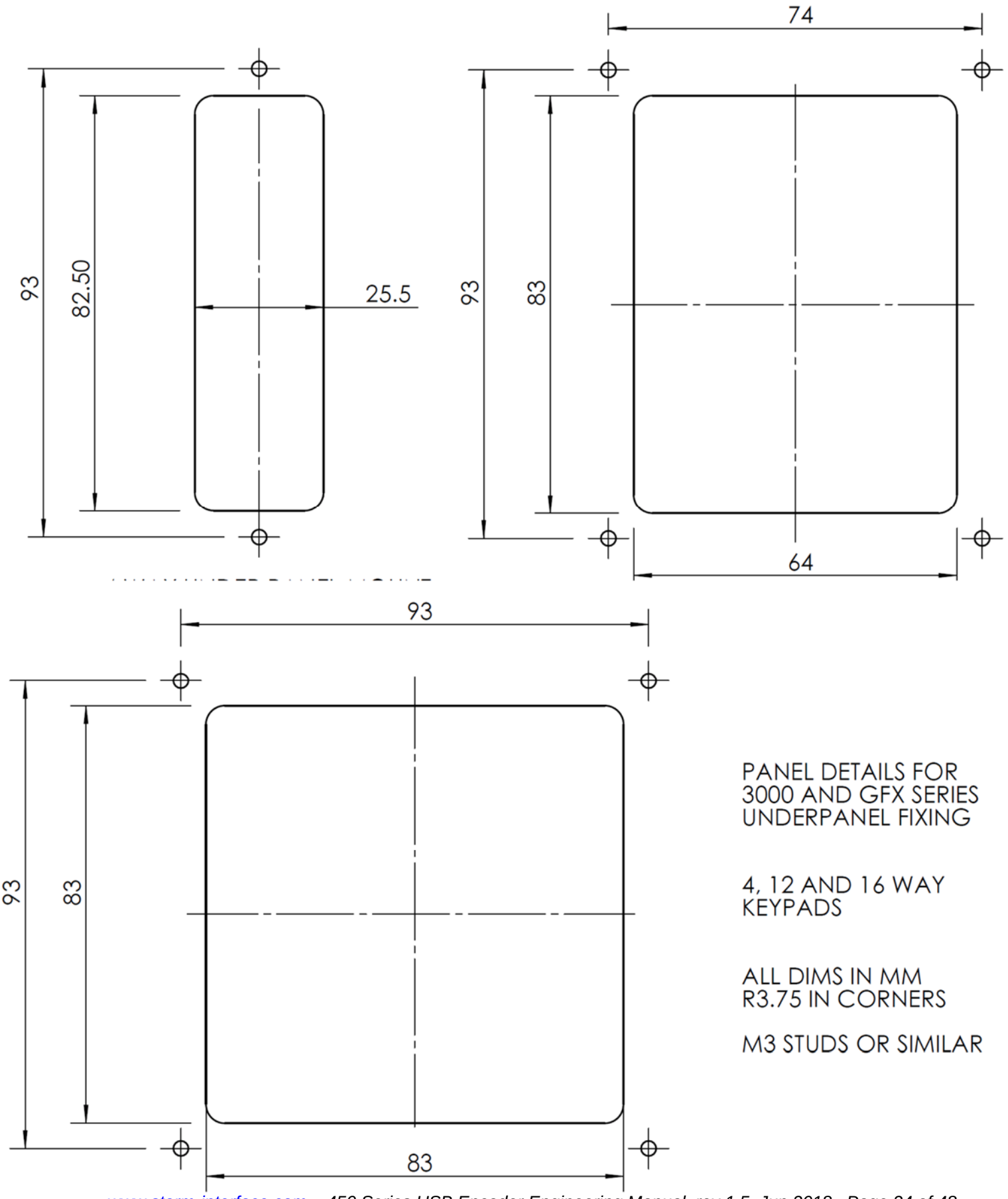
Panel Cutout Drawings

Surface Fixing of Keypads : 3000 Series, GFX Series



Panel Cutout Drawings

Underpanel Fixing of Keypads : 3000 Series, GFX Series,



Copyright Notice

This document is provided for use and guidance of engineering personnel engaged in the installation or application of Storm Interface data entry products manufactured by Keymat Technology Ltd. Please be advised that all information, data and illustrations contained within this document remain the exclusive property of Keymat Technology Ltd. and are provided for the express and exclusive use as described above.

This document is not supported by Keymat Technology's engineering change note, revision or reissue system. Data contained within this document is subject to periodic revision, reissue or withdrawal. Whilst every effort is made to ensure the information, data and illustrations are correct at the time of publication, Keymat Technology Ltd. are not responsible for any errors or omissions contained within this document.

No part of this document may be reproduced in any form or by any means or used to make any derivative work (such as translation or adaptation) without written permission from Keymat Technology Ltd.

For more information about Storm Interface and its products, please visit our website at www.storm-interface.com

==========

Copyright Acknowledgement

This product uses binary format of the hidapi dll, Copyright (c) 2010, Alan Ott, Signal 11 Software. All rights reserved.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Change History

USB Configuration Utility	Date	Version	Details
4500-SW01	1 Aug 13	2.1	First Release
	20 Aug 13	3.0	Increased size of modifier button + Increased size of Select Code Combo box.
	12 Nov 13	4.0	Update in line with 8v04 release

USB Encoder Firmware	Date	Version	Details	Class
	1 Aug 13	8v02	First Release	
	20 Aug 13	8v03	Disable USB stack serialisation	
	12 Nov 13	8v04	Improve Brightness Control.	
	03 Mar 17	8v05	Improve stability	
	13 Dec 17	8v06	Improved recovery process, if USB state is in other state then enumerated.	
	20 Jan 18	8v07	Windows 10 update. Recovery from MCU going to sleep mode. This was part of USB recovery process.	Red for Win10 users
	30 May 18	8v08	Disable low power mode (this caused occasional missed key presses when MCU is waking up from suspend state)	Amber

Firmware Change Classification : Green = Minor Admin Change only
Amber = Recommend upgrade units in use
Red = Critical Update for units in use (possibly just for certain Platform / OS

Engineering Manual	Date	Version	Details
	1 Aug 13	1.0	First Release
	12 Aug 13	1.02	p7 Alternative Code table : Outputs with Numlock clarified. Also changed in French version p11-14. Checked some USB codes in the full tables. Removed Insulation Breakdown spec (error)
	1 Oct 2013	1.03	Add section about the API
	12 Nov 13	1.05	Software update to 8v04
	2 Nov 15	1.1	API added, plus addition of LED & buzzer control in API command set.
	6 May 2016	1.2	Fixed Molex pn ref for connection cable p17
	13 Feb 17	1.3	Added info for ribbon cables for remote install, software update to 8v05.
	22 Jun 17	1.4	added SF6000 series pinout connection to page 17
	1 Jun 18	1.5	Added firmware update notes.

API Documentation	Date	Version	Details
	1 Oct 2013	1.0	First Release
	2 Nov 15	API Doc merged with Engineering Manual	

API

USB Encoder Device Communications

The encoder 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 nothing is received.
When encoder transmits a key press code to host, it will only send ASCII code. No acknowledgement is required.

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 Format Example – enable the buzzer on key press

	STX	ID	DATA LENGTH	DATA FIELD	ETX	LRC
HEX	0x02	04	1	1	0x03	calculated

The full message is always as per the format above. For clarity only the ID and the Data Field are shown in the message definitions on pages 4 – 14.

Character Types Used

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

Message Definitions

The messages are listed below; each one is fully detailed on the following pages.

ID.	Data	Message	Description
01		Device Status Request	Host To USB Encoder – Output the firmware version and all currently selected parameters
02	lb	LED Brightness	Host To USB Encoder – adjust led brightness. (default: 0)
03		Reserved	
04	bof	Key Press Buzzer On/Off	Host To USB Encoder - Enable/Disable buzzer. (Default: Enable)
05	lt	Load New code table	Host To USB Encoder – Load new code table
06	bp	Change Buzzer Duration	Host To USB Encoder – change the buzzer period duration
07	kt	Keypad Type	Host To USB Encoder – Select layout table 0 – Function key – 4way (default) 1 – Arrow Key - 4 way 2 – Customised 4way, 3 – Telephone – 12way, 4 – Calculator – 12way , 5 – Customised – 12way, 6 – Telephone – 16way, 7 – Calculator – 16way , 8 – Customised – 16way
08	st	Self Test	Host To USB – The encoder start/end a self test
09		Save Configuration	Host To USB Encoder – Encoder writes configuration data from ram to flash.
10		Reset to factory default	Host To USB Encoder – Reset device back to factory default
11		Reserved	
12		Load Firmware	Host To USB Encoder – Sets the encoder to detect the device loader for firmware loading
13	sb	Status Buzzer	Host to USB encoder – Sounds the buzzer for x period. X is passed in value (0 -9)

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 ID

Following table shows the possible values for the device ID field:

00	Keymat Technology USB Encoder

Device Status (01)

Host sends this message to USB encoder to request the status of the encoder.

USB Encoder Status Response

USB Encoder sends this message to Host in response to the Device Status message.

	Data Field	Type	Length	Description
ec	Error Code	H	2	
ldb	LED Brightness	N	1	Value (0 – 9)
	reserved	N	1	
bof	Buzzer	N	1	0 – OFF, 1 – ON
op	Option	N	1	0x01 – LEDs, 0x02 – Buzzer, Rest is reserved for future use.
bp	Buzzer on period	N	1	Value (0 – 9)
kt	Keypad Type	N	1	0 – Function key – (default) 4 way 1 – Arrow Key 4 way 2 – Customised 4 way 3 – Telephone 12 way 4 – Calculator 12 way 5 – Customised 12 way 6 – Telephone 16 way 7 – Calculator 16 way 8 – Customised 16 way
ktab	Keycode table	H	Up to 32	Layout selected table – data could be for 4 way or 12/16 way
fwv	Firmware Version	A,N	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 “

Host Device		USB Encoder
[01]	⇒	
	⇐	[01][ec][ldb][bof][op][bp][kt][ktab][fwv]

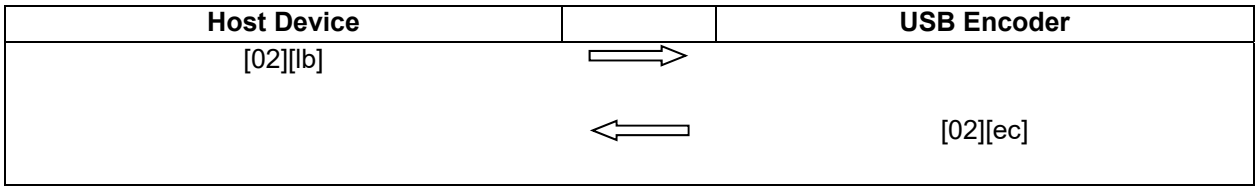
LED Brightness Command (02)

Host sends this message to control brightness of LEDs

	Data Field	Type	Length	Description
lb	LED brightness	N	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

Buzzer On / Off Command (04)

Host sends this message to enable/disable buzzer on key presses

	Data Field	Type	Length	Description
bof	Buzzer	N	1	0-Disable, 1-Enable

Buzzer Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		USB Encoder
[04][bof]	⇒	
	⇐	[04][ec]

Load New Key Code Table Command (05)

Host sends this message to Load New Code Table

	Data Field	Type	Length	Description
lt	Load New Code Table	H	Always 32	Key Code Table: 8 for 4W, 24 for 12W, or 32 for 16W

Load New Table Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		USB Encoder
[05][lt][32 scan codes]	⇒	
	⇐	[05][ec]

Note: Length is always 32, for example for 4 way, it requires 8 codes and this will be the first 8 bytes, subsequent 24 bytes will be ignored by the encoder.

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>< 0x00><0x00>

Note: 32 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

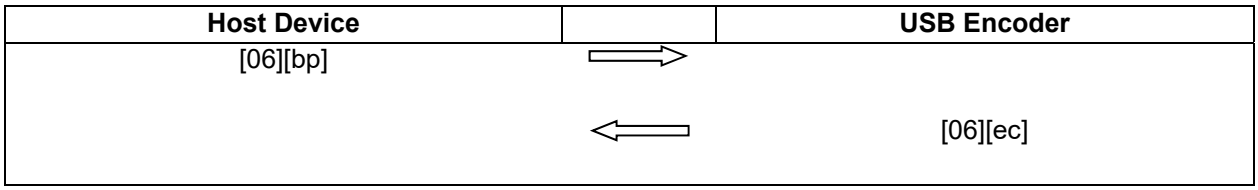
Buzzer Duration Command (06)

Host sends this message to change the duration of the buzzer period (when a key is pressed)

	Data Field	Type	Length	Description
bp	Duration	N	1	Value 0 - 9

Buzzer Duration Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	



Buzzer value increments in 0.25s. For example 1 = 0.25s, 2 – 0.5s, 3 – 0.75s, 4 – 1.0s etc.,

Keypad Type Command (07)

Host sends this message to set keypad type

	Data Field	Type	Length	Description
kt	Keypad Type	N	1	0 – Function key – (default) 4 way 1 – Arrow Key 4 way 2 – Customised 4 way 3 – Telephone 12 way 4 – Calculator 12 way 5 – Customised 12 way 6 – Telephone 16 way 7 – Calculator 16 way 8 – Customised 16 way

Keypad Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		USB Encoder
[07][kt]	⇒	
	⇐	[07][ec]

Self Test Command (08)

Host sends this command to request the USB Encoder to start/end a self test of the encoder.

	Data Field	Type	Length	Description
st	Self test	N	1	1 – start self test 2– end self test

Self Test Start/End Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	

Host Device		USB Encoder
[08][st]	⇒	
	⇐	[08][ec] and if error_code is 00 Then it will be followed by keypresses
	⇐	[08][keypresses] After self test is started and If keypad key is not pressed within 5 sec It will timeout and self test will end and get following [08][0xff]

The self test command is used to test the key presses on keypad.

Host USB Encoder

-------Self Test (1) ----------→

This command disables the USB encoder USB devices, so if any keys are pressed the key codes are sent to Host over the HID datapipe channel as shown above. So if user presses key button 1.

←-------[08][0] ---------- User presses Key 1

←------[08][01]---------- User presses Key 2

If no key is presses for 5 sec then the command ends

←-------[08][0xff]----------

The host can stop the self test command by issuing

----------Self test(2) ----------→

←----------[08][00]----------

Write Config Data To Flash command (09)

Host sends this command to request the USB Encoder 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	

Host Device		USB Encoder
[09]	⇒	
	⇐	[09][ec]

Reset To Factory Default command (10)

Host sends this command to request the USB Encoder 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		USB Encoder
[10]	⇒	
	⇐	[10][ec]

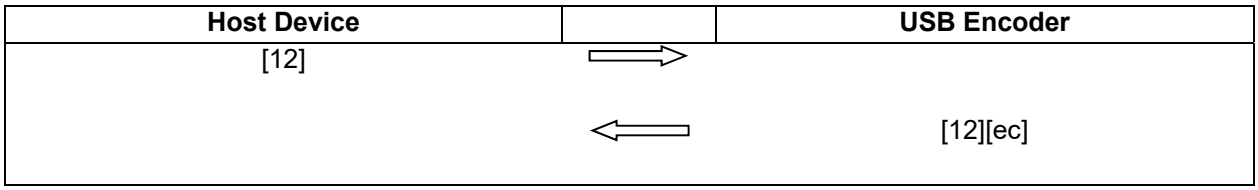
Enable BSL Command (12)

Host sends this command to request the USB Encoder to start downloader.

This command has no data associated with it.

Enable BSL Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	



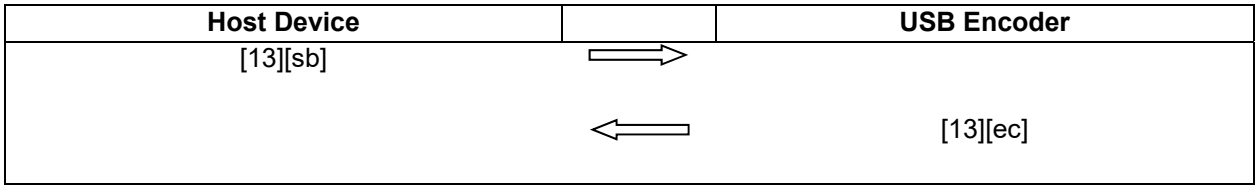
Status Buzzer Command (13)

Host sends this message to sound the buzzer for specified duration

	Data Field	Type	Length	Description
sb	Duration	N	1	Value 0 - 9

Status Buzzer Command Response

	Data Field	Type	Length	Description
ec	Error Code	H	2	



Key Press Code

Each time a key is pressed on keypad the USB encoder sends the keyboard report back to HOST.

When the key is released the USB Encoder sends 00000000 back to the HOST.

Keypress Code Type Report

	Data Field	Type	Length	Description
1	Key Press Code	A	1	Sends appropriate key code to host when keypad key is pressed.

Host Device		USB Encoder
	⇦	[KEY CODE]

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 a 4 way keypad connected to encoder and configured for Arrow key. If the user now presses the top key, which is “up arrow” and USB code of 52. Then keyboard report sent to host would be:

Byte 0 – 0
Byte 1 – 0
Byte 2 – 52
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 – 52
Byte 3 – 0
Byte 4 – 0
Byte 5 – 0
Byte 6 – 0
Byte 7 – 0

API Overview (Host Implementation)

The USB Encoder API Library is a library program which currently is tested on Windows (from XP and above) and Linux (Ubuntu) platform.

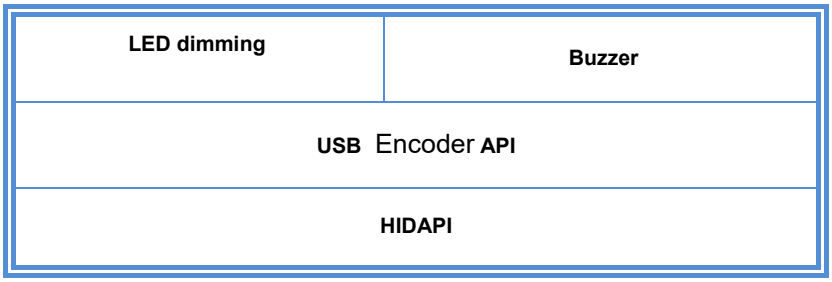
The Library is a middleware program between operating system and host application. The library encapsulates all the communication protocol and exposes a very simple API for host application.

This document is prepared for application developers who will implement a host application for the USB Encoder.

The USB Encoder API Library is a middleware application between USB Encoder Host application and USB Encoder system.

The USB Encoder uses USB for communicating with the host. It includes an HID-compliant device . One of the advantages of using this implementation, which using 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 of the USB Encoder API is shown below.



- USB Encoder API – The USBEncoderApi library allows for the host application to invoke USB Encoder 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.

Libraries are provided for both the HIDAPI and USB Encoder interface, so that it can be linked into the users host application. This exposes a well defined API for the host application.

The developer does not need to worry about the communication at low level. You can request source code for the implementation for 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

This is referenced in below functions:

```
  enum REQUEST_TYPE
  {
// message types
  DEVICE_STATUS = 1,            ///Device status message
  LED_BRIGHTNESS,
  LED_COLOUR,
  BUZZER_ON_OFF,
  LOAD_KEYCODE_TABLE,
  BUZZER_PERIOD,
  KEYPAD_TYPE,
  RESERVED_1,
  WRITE_CONFIG,
  FACTORY_DEFAULT,
  RESERVED_2,
  FIRMWARE_LOAD,
  STATUS_BUZZER
};
```

InitialiseStormUSBDevice

This function is used to initialise the USB Encoder. The usb encoder is identified by the Product PID and Manufacturer VID. These are assigned to Keymat:

- Vendor ID – 0x2047
- Product ID – 0x0902

On successful finding the USB Encoder the manufacturer_local will be filled with “Storm Interface” and product_local will be filled with “USB Encoder”. If not successful both of the strings will be filled with “none”

Parameters :

storm_vid	-	Vendor ID
product_pid	-	Product ID
manufacturer_local	-	vendors name will be stored
product_local	-	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, std::string
&manufacturer_local, std::string &product_local );


or


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

Get Device Status

This function retrieves status information about the USB encoder. For example, LED status etc. All information is stored in DEVICE_INFO structure.

```
typedef struct
{
      unsigned char           led_brightness;
      unsigned char           led_color;
      unsigned char           buzzer;
      unsigned char           buzzer_period;
      unsigned char           keypad_type;
      unsigned char           keypad_table[32];
      std::string             FirmwareName;

} DEVICE_INFO;



///   \brief GetDeviceStatus Retrieves the keypad's status and information including:
///   Serial Number, Tamper Status, Firmware Version, 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 450 Encoder

///   \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_ENCODER_CONNECTED = No keypad is connected so cannot retrieve info

///   REQUEST_TIMEOUT   = Could not retrieve the info in the time alloted.
///

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

Set Led Brightness

This function sets the led brightness. The brightness can be set from level 0 to 9

0 will set the leds to off

```
///\brief SetLEDBrightness : Sets the LED brightness between 0 to 9, where 0 is off
///\param led_brightness             0 to 9, where 0 is off
///\return  0 on success, negative error code on failure
      //

int SetLEDBrightness(  int led_brightness,   int _timeToWait );
```

Enable/Disable Buzzer

```
This function will enable or disable the buzzer.

Buzzer = 1  enable
Buzzer = 0 Disable

///\brief EnableDisableBuzzer : Enables/Disables buzzer
      ///\param buzzer         0-disable, 1-enables
      ///\return  0 on success, negative error code on failure
      //

int EnableDisableBuzzer(  int buzzer,   int _timeToWait );
```

Beep Buzzer

This function will sound the buzzer for the buzzer_period. The buzzer period is defined between 0 - 9

```
///\brief SetBuzzerPeriod : Sets buzzer period betweeen 0 -9
      ///\param buzzer_period        0-off, 1 - 9
      ///\return  0 on success, negative error code on failure
      //

int   SetBuzzerPeriod(  int buzzer_period,   int _timeToWait );
```