GRAPHITI®

API Specification

7th July 2021
Version 0.22

Contents

1 Introduction

This document describes the communication protocol between a host device (a PC, for instance) and the Graphiti® Tactile Graphics Display, with detailed descriptions of each command.

You can create your own applications on a platform of your choice to communicate with the Graphiti using this protocol.

2 Interface Overview

The Graphiti device is accessible using its USB port (via HID or VCP) or over Bluetooth (via the SPP profile). The host communicates with the Graphiti device using commands described in the following sections. The device will send an acknowledgement for every command received from the host.

2.1 *Selecting the interface*

On the Graphiti device, the physical interface can be selected using the following commands on the keypad.

Interface	Command
USB HID	SPACE + DOT 7
USB VCP (Serial)	SPACE + DOT 8
Bluetooth (SPP)	SPACE + DOT 4

2.2 *Device information*

#	Field	Value
1	Product Name	“Graphiti”
2	USB Vendor ID (VID) and Product ID (PID) for HID and VCP	VID: 0x1FC9, PID: 0x8217
3	Bluetooth Name	Same as the unit serial number
4	Size of the display	60x40 pixels
5	Communication modes supported	USB HID, USB VCP (Serial), Bluetooth (SPP profile)

2.3 *Interfacing using serial communication software*

Any software that supports serial communication can be used to communicate with the device through the virtual communication port generated in the host device.

Such software is readily available or can be created by you. Some examples include:

- Hyperterminal
- Terminal
- Docklight

The serial mode configuration parameters are as shown below:

Parameter	Value
Baud Rate	115200
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	None / Off

After setting these configuration parameters, select and open the virtual communication port in your host software. You can now begin sending commands from the command set described below.

3 General Specifications

- Command (Host to Graphiti)
  You need to follow the exact command syntax while transmitting data from the host to the device.
- Response (Graphiti to Host)
  The device will either send a response or an error message back to the host on receiving or processing the message received from the Host.
- Start of frame is indicated by 0x1B (ESC) in VCP mode. In HID mode, one more byte (i.e., Report ID) is shown at the start of frame (0x1B (ESC)).
- In a frame, the byte after SOF indicates the command ID.
- If a data byte is the same as SOF (0x1B), you will have to send it twice to indicate that byte as data. The device will also send the data byte twice if it is the same as SOF. Please note that this rule is not applicable to the 'Send Image (Blocking)' API. Also, do not include the extra byte for the checksum calculation.
- The device will flush the previous action if a new command is received from the host. A host must send ACK or NACK to the device on receiving a response from the device.
- You must use modular sum method to calculate the checksum value. i.e. add all the bytes as unsigned binary numbers, discard any overflow bits, and append the two's complement of the total as the checksum.
  Note that the report ID (in HID) and SOF should not be considered while calculating checksum.
- If the checksum byte is set to 0x00, the device will ignore the checksum validation.
- If the checksum value is the same as SOF (0x1B), you must send it twice to indicate it as data.

4 Commands for USB VCP (Serial) or Bluetooth (SPP) interface

4.1 *ACK and NACK request*

ACK command: You can send this command to the device to inform that your received data is correct.
NACK command: Whenever the checksum error is detected in the received data at the host side, you can send NACK to the device so that the device will send the same data again. The host application must send NACK command within 300ms of the previous response, otherwise the device will not serve it.
The ACK/NACK request should be sent to the device, only after receiving a response to the commands.

ACK syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x51	0xAF

NACK syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x52	0xAE

4.2 *Graphiti Information Commands*

4.2.1 Get Software Version

The ‘Get Software Version’ command retrieves the version of the current firmware in the device.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x01	0xFF

Response:

Byte 0	Byte 1	Byte 2-16	Byte 17
0x1B	0x01	Software Version (ASCII bytes, 15 bytes, fixed length)	Checksum

4.2.2 Get Hardware Version

The ‘Get Hardware Version’ command retrieves the current version of the device hardware.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x02	0xFE

Response:

Byte 0	Byte 1	Byte 2-16	Byte 17

0x1B	0x02	Hardware Version (ASCII bytes, 15 bytes, fixed length)	Checksum

4.2.3 Get Unit Serial Number

The 'Get Unit Serial Number' command retrieves the serial number of the connected device.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x04	0xFC

Response:

Byte 0	Byte 1	Byte 2-16	Byte 17
0x1B	0x04	Unit Serial Number (ASCII bytes, 15 bytes, fixed length)	Checksum

4.2.4 Get Battery Status

The 'Get Battery Status' command retrieves the current battery state of the device including the charging status.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x05	0xFB

Response:

Byte 0	Byte 1	Byte 2-16	Byte 17
0x1B	0x05	Battery Charging Status **Example: "**Charging 50%"/ "Battery 50%" (ASCII bytes, 15 bytes, fixed length)	Checksum

4.2.5 Get Resolution Information

The 'Get Resolution Information' command retrieves the details of horizontal and vertical resolution supported by the device.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x07	0xF9

Response:

Byte 0	Byte 1	Byte 2-3	Byte 4	Byte 5-6	Byte 7 (Checksum)
0x1B	0x07	Horizontal Resolution (ASCII 2 bytes)	Separator ('|')	Vertical Resolution (ASCII 2 bytes)	0xB3

4.2.6 Get Device Orientation

The ‘Get Device Orientation’ command retrieves the current orientation of the device. Presently the device supports landscape orientation only.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x08	0xF8

Response:

Byte 0	Byte 1	Byte 2-11	Byte 12 (Checksum)
0x1B	0x08	The Device Orientation “Portrait” or “Landscape” (ASCII, 10 bytes, fixed length)	0x6D

Note: The device will return only “Landscape” orientation because “Portrait” orientation is not supported.

4.2.7 Get Height Information

The ‘Get Height Information’ command retrieves information about the number of height levels supported by the device for all the pins.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x09	0xF7

Response:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x09	‘4’ = supports four different height levels	0xC3

4.2.8 Get Device Name

The ‘Get Device Name’ command is used to retrieve the name of the device.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x0A	0xF6

Response:

Byte 0	Byte 1	Byte 2-31	Byte 32 (Checksum)
0x1B	0x0A	The Device Name “Graphiti” (ASCII bytes, 30 bytes, fixed length)	0xBE

4.3 *Display Access Commands*

The response to all the commands in this category is received only after the display has been updated.

Information regarding the height and blink rate is shown in the table below. This information is useful while using display access commands with blink feature.

Pixel Value	Pixel Height
0x00	No height
0x01	Height level one
0x02	Height level two
0x03	Height level three
0x04	Height level four

Blinking Value	Blinking Rate
0	No Blinking
1	100ms
2	200ms
3	300ms
…	…
50	5secs

4.3.1 Update Display

The 'Update Display' command is used to configure the pins to the required height.

For Graphiti, each pixel value and its blinking value is represented by one byte each. Therefore, to update the entire display of 40 x 60 pixels, it is necessary to transmit command with the value of each pixel and its blinking rate as represented in the syntax below. The second byte represents the first pixel position on the display array and the third byte represents the blinking value for first pixel. Similarly, 4800th byte represents the 2400th pixel on the display array and 4801st byte represents 2400th pixel's blinking value.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	…	Byte 4800	Byte 4801	Byte 4802
0x1B	0x15	1st pixel value (Decimal 0 to 4)	1st pixel blink rate	…	2400th pixel value (Decimal 0 to 4)	2400th pixel blinking rate	Checksum

Position of pins on the display:
First line bytes: 1 to 60
Last line bytes: 2340 to 2400

For example,
The command shown below will update the complete display as per the input height level of each pixel and blink only the first pixel once in 500ms.
0x1B 0x15 0x04 0x05 0x04 0x00 0x04 0x00 0x04 0x00 0x04 … (for 2400pixels) checksum (last byte)

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

4.3.2 Set / Clear Display

The 'Set / Clear Display' command is used to set or reset all the pins of the display to the highest height or lowest (in line with the surface) position, respectively.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x16	Set / Clear Display 0x02 = Set Display, 0x03 = Clear Display	0xE8: for Set Display 0xE7: for Clear Display

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

4.3.3 Update Single pixel on Display

The 'Update Single Pixel' command is used to configure a single pin to the desired height position and blink rate interval.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x1B	0x17	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Pixel value (Decimal 0 to 4)	Blinking rate	Checksum

Display view:

	Column ID (1 to 60)												
Row ID (1 to 40)	1	2	.	.	.	.	.	.	.	.	.	.	60
	2												
	.												
	.												
	.												
	40												

For example:

The command shown below will update the pixel in the 20th row and 31st column to height level 4.
0x1B 0x17 020 (decimal) 031 (decimal) 0x04 0x00 checksum

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

4.3.4 Update Single Row on Display

The 'Update Single Row' command is used to configure a single row to a desired height position. To update the entire row, the pixel value and blinking rate for each pixel in a row (60 pixels) needs to be provided.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	…	Byte 121	Byte 122	Byte 123
0x1B	0x18	Row ID (Decimal 1 to 40)	1st pixel value (Decimal 0 to 4)	1st pixel blinking rate	…	60th pixel value (Decimal 0 to 4)	60th pixel blinking rate	Checksum

For example:
The command below will update the 15th row to height level 4 without blinking.
0x1B 0x18 015(decimal) 0x04 0x00 0x04 0x00 … checksum.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

4.3.5 Update Single Column on Display

The 'Update Single Column' command is used to configure a single column to a desired height position. To update the entire column, the pixel value and blinking rate for each pixel in a column (40 pixels) needs to be provided.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	…	Byte 81	Byte 82	Byte 83
0x1B	0x19	Column ID (Decimal 1 to 60)	1st pixel value (Decimal 0 to 4)	1st pixel blinking rate	…	40th pixel value (Decimal 0 to 4)	40th pixel blinking rate	Checksum

For example:
The command shown below will update the 20th column to height level 4 without blinking.
0x1B 0x18 020 (decimal) 0x04 0x00 0x04 0x00… checksum.

Response:
The device will send a success or error response as mentioned in the section “Common Device Responses”.

4.3.6 Get All Pixels’ Position Status

The ‘Get All Pixels Position Status’ command is used to retrieve the present status (position) of each pixel of the entire display (40x60) in a single response.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x20	0xE0

Response:

Byte 0	Byte 1	Byte 2-2401	Byte 2402
0x1B	0x20	d	Checksum

Note: d represents the value of each pixel and the value of ‘d' must be from 0 to 4 height levels.

The response for the entire display is indicated as one byte for each pixel. For example, if Byte 2 of the response is 0x04, it indicates that the first pixel is at height level 4 and so on.

4.3.7 Get Single Pixel Position Status

The ‘Get Single Pixel Position Status’ command is used to retrieve the present status of the requested pixel.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x1B	0x21	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x1B	0x21	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	d	Checksum

Note: d represents the value of the pixel (Height Levels = 0, 1, 2, 3, 4)
The height of requested pixel is provided at the fourth byte of the response.

4.3.8 Get Single Row Pixels Position Status

The ‘Get Single Row Pixels Position Status’ command is used to retrieve the present status of each pixel in the requested row.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3
0x1B	0x22	Row ID (Decimal 1 to 40)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3-62	Byte 63
0x1B	0x22	Row ID (Decimal 1 to 40)	d	Checksum

Note: d represents the value of each pixel, and the value of d must be from 0 to 4 height Levels.

The response for single requested row is indicated as one byte for each pixel. For example, if Byte 3 of the response is 0x04, it indicates that the first pixel of the requested row is at height level 4 and so on.

4.3.9 Get Single Column Pixels Position Status

The 'Get Single Column Pixels Position Status' command is used to retrieve the present status of each pixel in the requested column.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3
0x1B	0x23	Column ID (Decimal 1 to 60)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3-42	Byte 43
0x1B	0x23	Column ID (Decimal 1 to 60)	d	Checksum

Note: d represents the value of each pixel, and the value of d must be from 0 to 4 height Levels.

The response for single requested column is indicated as one byte for each pixel. For example, if Byte 3 of the response is 0x04, it indicates that the first pixel of the requested column is at height level 4 and so on.

4.3.10 Show Message on the Device

The 'Show Message' command is used to show the text message on the display.

Syntax:

Byte0	Byte1	Byte2	Byte 3	…	Byte40	Byte 41	Byte42	Byte 43
0x1B	0x24	1st Character of Message	1st Character Select Flag	…	20th Character of Message	20th Character Select Flag	Cursor Blinking Rate	Checksum

Here,
When the byte for select flag is set to 1, it will underline the respective character.
When the byte for select flag is set to 2, it will display the cursor at respective character.

Response:

The device will send a success or error response as mentioned in the section “Common Device Responses”.

4.3.11 Set Cursor

The ‘Set Cursor’ command is used to set the cursor on the display. Here, you need to give the cursor position and size of the cursor.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x1B	0x25	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Cursor height (Decimal 1 to 4)	Cursor length (Decimal 1 to 60)	Cursor blinking rate	Checksum

The cursor will blink at the rate which is equal to Cursor blinking rate value multiplied by 100(milliseconds).

The cursor height indicates the pin height level, i.e., if you set the cursor at height level 2, that pixel will blink up to height level 2.

Response:
The device will send a success or error response as mentioned in the section “Common Device Responses”.

4.3.12 Send Image (Interruptible)

This command is used to send and display an image file, such as a BMP, PNG, or JPG file on the device. It is possible to terminate the data transfer of an image in this API and send a new image.

The file name must be include the file extension, for example ‘map.png’. The host must send the vertical bar symbol (|) as a separator after the file name.

In the image, if a data byte is the same as SOF (0x1B), send it twice to represent that byte as a data and while calculating checksum do not consider this extra byte. Like other APIs, do not include the first byte i.e., SOF in the checksum calculation. Please note that if checksum byte is 0x1B, it is necessary to send it twice.

The device gives a single short vibration every 2 seconds after the image reception starts, until the image is displayed on the device.

In case of failure, the device will wait for 3 seconds and give you an error response. Once the image is displayed on the device, you can save it to storage media by pressing “**Space + s**” keys on the device.

Syntax:

Byte 0	Byte 1	Byte varies between 2 to 257	Separator after file name	4 Byte after file name separator (n to n+4)	Byte (n+5) onwards	…	Byte N
0x1B	0x2F	Name of the Image (up to 255 Bytes) including file extension	'|' (Vertical bar symbol)	Image size (4 bytes, MSB is 1st Byte and LSB is 4th Byte after separator)	Image data including header (binary form)		Check sum

Response:
The device will send success or error responses as mentioned in the section "Common Device Responses".

4.3.13 Send Image (Blocking)

This command is used to display the actual image such as a BMP, PNG, or JPG file on the device.

The image API is different from the other APIs where checking for 0x1B is not required for the image data. Here, you must use the image data as it is, without modifying it and will also have to provide the actual image size.

The file name must be with the file extension, for example 'map.png'. The host must send vertical bar symbol (|) as a separator after Image name. The checksum calculation is the same as the other APIs where the first byte needs to be ignored i.e., SOF. Please note that if checksum byte is 0x1B, it is necessary to send it twice.

Once the host has initiated data transmission, it must send all data bytes as per the size mentioned in the "Image size" field. The device produces a single short vibration at every 2 seconds after the image reception starts until the image is displayed on the device.

In case of failure, the device will wait for 3 seconds and give you an error response. Once the image is displayed on the device, you can save it to storage media by pressing "**Space + s**" keys on the device.

Syntax:

Byte 0	Byte 1	Byte varies between 2 to 257	Separator after file name	4 Byte after file name separator (n to n+4)	Byte (n+5) onwards	…	Byte N

0x1B	0x30	Name of the Image (up to 255 Bytes) including file extension	'|' (Vertical bar symbol)	Image size (4 bytes, MSB is 1st Byte and LSB is 4th Byte after separator)	Image data including header (binary form)		Check sum

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

4.4 *User Keys Access Commands*

4.4.1 Set Key Event

The 'Set Key Event' command is used to enable or disable the key press event. When enabled, you will be able to get the information of each key press of the device keypad. This information will include the key value and its event type.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x31	Update Key Event 0x00 = Disable, 0x01 = Enable	0xCF: Disable 0xCE: Enable

Responses:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x31	Key Event Status after update 0x00 = Disabled, 0x01 = Enabled	0xCF: Disable 0xCE: Enable

If enabled, on each key press, you will receive the following response without any request from the host.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x1B	0x32	Key value (Higher Byte)	Key value (Lower Byte)	Key Press Type (0x02 = Short Press)	Checksum

Note: Presently the Graphiti supports short key press only.

The key values sent by the device for different keys are shown in the table below.

Key	Key Value
DOT 1	0x1000
DOT 2	0x2000
DOT 3	0x4000

DOT 7	0x8000
DOT 4	0x0100
DOT 5	0x0200
DOT 6	0x0400
DOT 8	0x0800
Up	0x0010
Left	0x0020
Down	0x0040
Right	0x0080
Select	0x0001
Space	0x0002

If multiple keys are pressed together, the device will send the sum of the values of the pressed keys. For example, if DOT1 and DOT 4 keys are pressed together, the device will send a final value 0x1100. At the host side, bit level operations can help to decode the exact key press from the multiple key input.

Key positions:

DOT 3	DOT2	DOT1		Up		DOT 4	DOT 5	DOT 6
			Left	Select	Right			
				Down				
			DOT 7	Space	DOT8			

Reserved key events:

Key event	Function
DOT 5 + DOT 6 + DOT 8	Firmware upgrade
DOT 8 + Down	Hard reset
DOT 7 + DOT 8	Unregister key events from the device

4.5 *Touch Access Commands*

4.5.1 Set Touch Event

The ‘Set Touch Event’ command is used to enable or disable the touch event. When this API is enabled, you will be able to get information of each touch event (after enabling the respective mode: either gesture mode or draw mode) on the touch panel.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x41	Set Touch Event 0x00 = Disable, 0x01 = Enable	0xBF: Disable 0xBE: Enable

Response:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)

0x1B	0x41	Touch Event Status after update 0x00 = Disabled, 0x01 = Enabled	0xBF: Disable 0xBE: Enable

After enabling touch event or gesture mode, on each valid gesture, you will receive the following response without any request from the host.

Byte 0	Byte 1	Byte 2	Byte 3
0x1B	0x42	Gesture ID	Checksum

The gesture corresponding to gesture id is described in the table below:

Gesture	**Gesture ID (In decimal)**
Double Tap	2
Down Swipe	3
Up Swipe	4
Right Swipe	5
Left Swipe	6

If the draw feature is enabled along with the touch event, you will receive the following response on each touch (for drawing) without any request from the host.

Byte 2 (Length) gives the total length of data (excluding the start of frame, command ID and checksum byte). After the length byte is the row ID, column ID, pin height and blink rate of the location where touch has been made.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	…	Byte n	Byte n+1	Byte n+2	Byte n+3	Byte n+4
0x1B	0x43	Length	Pin Row ID	Pin Column ID	Pin Height	Pin Blinking rate		Pin Row ID	Pin Column ID	Pin Height	Pin Blinking rate	Checksum

If the display is cleared in draw mode, the device will send the following response.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x43	0x00	0xBD

Note: The device will send gestures ID in gesture mode only and the device will send drawing details in draw mode only.

4.5.2 Get Last Touch Point Status

The ‘Get Last Touch Point Status’ is used to get the height of the pin which was last touched. To get last touch point status, you must first touch the pin and then transmit command from the host.

Note: This API will work in touch review mode only.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x44	0xBC

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x1B	0x44	Pin Row ID	Pin Column ID	Pin Height	Checksum

4.6 *Vibrator Control Command*

This command is used to drive vibrators available in the device with different frequency, duty cycle and duration.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x1B	0x61	Frequency Range (10-100 kHz) (Hex)	Duty cycle in Percentage (40 to 100) (Hex)	Duration Higher Byte (100-1000ms) (Hex)	Duration Lower Byte (100-1000ms) (Hex)	Checksum

Response:

The device will send a success or error response as mentioned in the section “Common Device Responses”.

4.7 *Clock*

4.7.1 Get Date and Time

The ‘Get Date and Time’ command is used to get the date and time information of the device. The device provides clock information in the 24-hour format.

Syntax:

Byte 0	Byte 1	Byte 2 (Checksum)
0x1B	0x65	0xBB

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
0x1B	0x65	D (ASCII)	D (ASCII)	-	M (ASCII)	M (ASCII)	-	Y (ASCII)	Y (ASCII)

Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte15	Byte 16	Byte 17
Y (ASCII)	Y (ASCII)	Space (ASCII)	H (ASCII)	H (ASCII)	:	M (ASCII)	M (ASCII)

Byte18	Byte 19	Byte 20	Byte 21
:	S (ASCII)	S (ASCII)	Checksum

4.7.2 Set Date and Time

The 'Set Date and Time' command is used to set the device date and time in 24-hour format. The device will accept data in 24hr format.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x1B	0x66	DD (Hex)	MM (Hex)	YY (Higher byte of year in hex)	YY (Lower byte of year in hex)

Byte 6	Byte 7	Byte 8	Byte 9
HH (Hex)	MM (Hex)	SS (Hex)	Checksum

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5 Commands for USB HID interface

API commands for HID Mode are almost the same as those for VCP mode, except that they contain the report ID as the first byte.

5.1 *ACK and NACK request*

The ACK/NACK request should be sent to the device only after receiving a response to the command.

ACK syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x51	0xAF

NACK syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x52	0xAE

5.2 *Graphiti Information Commands*

5.2.1 Get Software Version

The 'Get Software Version' command retrieves the version of the current firmware in the device.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x01	0xFF

Response:

Byte 0	Byte 1	Byte 2	Byte 3-17	Byte 18
0x07	0x1B	0x01	Software Version (ASCII bytes, 15 bytes, fixed length)	Checksum

5.2.2 Get Hardware Version

The ‘Get Hardware Version’ command retrieves the current version of the device hardware.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x02	0xFE

Response:

Byte 0	Byte 1	Byte 2	Byte 3-17	Byte 18
0x07	0x1B	0x02	Hardware Version (ASCII bytes, 15 bytes, fixed length)	Checksum

5.2.3 Get Unit Serial Number

The ‘Get Unit Serial Number’ command retrieves the serial number of the connected device.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x04	0xFC

Response:

Byte 0	Byte 1	Byte 2	Byte 3-17	Byte 18
0x07	0x1B	0x04	Unit Serial Number (ASCII bytes, 15 bytes, fixed length)	Checksum

5.2.4 Get Battery Status

The ‘Get Battery Status’ command retrieves the current state of the device battery including the charging status.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x05	0xFB

Response:

Byte 0	Byte 1	Byte 2	Byte 3-17	Byte 18

0x07	0x1B	0x05	Battery Charging Status **Example: "**Charging 50%"/ "Battery 50%" (ASCII bytes, 15 bytes, fixed length)	Checksum

5.2.5 Get Resolution Information

The 'Get Resolution Information' command retrieves the details of the horizontal and vertical resolution supported in the device.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x07	0xF9

Response:

Byte 0	Byte 1	Byte 2	Byte 3-4	Byte 5	Byte 6-7	Byte 8 (Checksum)
0x07	0x1B	0x07	Horizontal Resolution (ASCII 2 bytes)	Separator ('|')	Vertical Resolution (ASCII 2 bytes)	0xB3

5.2.6 Get Device Orientation

The 'Get Device Orientation' command retrieves the current orientation of the device. Presently, the device supports landscape orientation only.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x08	0xF8

Response:

Byte 0	Byte 1	Byte 2	Byte 3-12	Byte 13 (Checksum)
0x07	0x1B	0x08	The Device Orientation "Portrait" or "Landscape" (ASCII maximum 10 bytes)	0x6D

Note: The device will return only "Landscape" orientation, because "Portrait" orientation is not supported.

5.2.7 Get Height Information

The 'Get Height Information' command retrieves information about the number of height levels supported by the device for all the pins

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x09	0xF7

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x09	'4' = supports four different height levels	0xC3

5.2.8 Get Device Name

The 'Get Device Name' command is used to retrieve the name of the device.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x0A	0xF6

Response:

Byte 0	Byte 1	Byte 2	Byte 3-32	Byte 33
0x07	0x1B	0x0A	The Device Name "Graphiti" (ASCII bytes, Maximum 30 bytes)	0xBE

5.3 *Display Access Commands*

The response for all the commands in this category is received after the display is updated.

Information regarding height and blink rate is shown below. You need to use this information while using display access commands with blink feature.

Pixel Value	Pixel Height
0x00	No height
0x01	Height level one
0x02	Height level two
0x03	Height level three
0x04	Height level four

Blinking Value	Blinking Rate
0	No Blinking
1	100ms
2	200ms
3	300ms
…	…
50	5secs

5.3.1 Update Display

The 'Update Display' command is used to configure the pins to the required height.

For Graphiti, each pixel value and its blinking value is represented by one byte each. Therefore, to update the entire display of 40 x 60 pixels, it is necessary to transmit the command with value of each pixel and its blinking rate as represented in the syntax below. Second byte represents the first pixel position on the display array and third byte represents the blinking value for the first pixel. Similarly, the 4801st byte represents the 2400th pixel on the display array and the 4802nd byte represents the 2400th pixel's blinking value.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	…	Byte 4801	Byte 4802	Byte 4803
0x03	0x1B	0x15	1st pixel value (Decimal 0 to 4)	1st pixel blink rate	…	2400th pixel value (Decimal 0 to 4)	2400thpixel blinking rate	Checksum

Position of pins on the display:
First line bytes: 1 to 60
Last line bytes: 2340 to 2400

For example:
The command shown below will update the complete display as per the input height level of each pixel and blink only the first pixel once in 500ms.
0x03 0x1B 0x15 0x04 0x05 0x04 0x00 0x04 0x00 0x04 0x00 0x04 … (for 2400 pixels) Checksum (last byte).

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.2 Set / Clear Display

The 'Set / Clear Display' command is used to set or reset all the pins of the display to the highest height or lowest (in line with the surface) position, respectively.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x02	0x1B	0x16	Set / Clear Display 0x02 = Set Display, 0x03 = Clear Display	0xE8: Set Display 0xE7: Clear Display

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.3 Update Single Pixel on Display

The 'Update Single Pixel' command is used to configure a single pin to a desired height position and blink rate interval.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x02	0x1B	0x17	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Pixel value (Decimal 0 to 4)	Blinking rate	Checksum

Display view:

	Column ID (1 to 60)											
Row ID (1 to 40)	1	2	.	.	.	.	.	.	.	.	.	60
	2											
	.											
	.											
	40											

For example:
The command shown below will update the pixel at 20th row and 31st column to the height level of 4.
0x02 0x1B 0x17 020(decimal) 031(decimal) 0x04 0x00 Checksum.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.4 Update Single Row on Display

The 'Update Single Row' command is used to configure a single row to a desired height position. To update the entire row, you need to provide pixel value and blinking rate for each pixel in a row (60 pixels).

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	…	Byte 122	Byte 123	Byte 124
0x03	0x1B	0x18	Row ID (Decimal 1 to 20)	1st pixel value (Decimal 0 to 4)	1st pixel blinking rate	…	60th pixel value (Decimal 0 to 4)	60th pixel blinking rate	Checksum

For example:

Below command will update the 15th row to the height level 4 without any blinking.
0x03 0x1B 0x18 015(decimal) 0x04 0x00 0x04 0x00 … Checksum.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.5 Update Single Column on Display

The 'Update Single Column' command is used to configure a single column to a desired height position. To update the entire column, you need to provide pixel value and blinking rate for each pixel in a column (40 pixels).

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	…	Byte 82	Byte 83	Byte 84
0x03	0x1B	0x19	Column ID (Decimal 1 to 32)	1st pixel value (Decimal 0 to 4)	1st pixel blinking rate	…	40th pixel value (Decimal 0 to 4)	40th pixel blinking rate	Checksum

For example:
The command shown below will update the 20th column to the height level 4 without blinking.
0x03 0x1B 0x18 020(decimal) 0x04 0x00 0x02 0x00 … Checksum.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.6 Get All Pixels Position Status

The 'Get All Pixels Position Status' command is used to retrieve the present status (position) of each pixel of the entire display (40 x 60) in a single response.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x20	0xE0

Response:

Byte 0	Byte 1	Byte 2	Byte 3-2402	Byte 2403
0x08	0x1B	0x20	d	Checksum

Note: d represents the value of each pixel, and the value of d must be from 0 to 4 height Levels.

The response for the entire display is indicated as one byte for each pixel. For example, if second byte of the response is 0x04, it indicates that the first pixel is at height level 4 and so on.

5.3.7 Get Single Pixel Position Status

The 'Get Single Pixel Position Status' command is used to retrieve the present status of the requested pixel.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x02	0x1B	0x21	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x07	0x1B	0x21	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	d	Checksum

Note: d represents the value of the pixel (Height Levels = 0, 1, 2, 3, 4)
The height level for the requested pixel is provided at fourth byte of the response.

5.3.8 Get Single Row Pixels Position Status

The 'Get Single Row Pixels Position Status' command is used to retrieve the present status of each pixel in the requested row.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x02	0x1B	0x22	Row ID (Decimal 1 to 40)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4-63	Byte 64
0x08	0x1B	0x22	Row ID (Decimal 1 to 40)	d	Checksum

Note: d represents the value of each pixel, and the value of d must be from 0 to 4 height Levels.

The response for single requested row is indicated as one byte for each pixel. For example, if the third byte of the response is 0x04, it indicates that the first pixel of the requested row is at height level 4 and so on.

5.3.9 Get Single Column Pixels Position Status

The 'Get Single Column Pixels Position Status' command is used to retrieve the present status of each pixel in the requested column.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x02	0x1B	0x23	Column ID (Decimal 1 to 60)	Checksum

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4-43	Byte 44
0x07	0x1B	0x23	Column ID (Decimal 1 to 60)	d	Checksum

Note: d represents the value of each pixel, and the value of d must be from 0 to 4 height Levels.

The response for single requested column is indicated as one byte for each pixel. For example, if the third byte of the frame response is 0x04, it indicates that the first pixel of the requested column is at height level 4 and so on.

5.3.10 Show Message on the Device

The 'Show Message' command is used to show the text message on the display.

Syntax:

Byte0	Byte1	Byte2	Byte 3	Byte 4	…	Byte 41	Byte 42	Byte 43	Byte 44
0x02	0x1B	0x24	1st Character of Message	1st Character Select Flag	…	20th Character of Message	20th Character Select Flag	Cursor Blinking Rate	Checksum

Here,
When select flag byte is set to 1, it will underline the respective character.
When select flag byte is set to 2, it will display the cursor at the respective character.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.3.11 Set cursor

The 'Set Cursor' command is used to set the cursor on the display. Here, you need to give the cursor position and size of the cursor.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x02	0x1B	0x25	Row ID (Decimal 1 to 40)	Column ID (Decimal 1 to 60)	Cursor height (Decimal 1 to 4)	Cursor length (Decimal 1 to 60)	Cursor blinking rate	Checksum

The cursor will blink at the rate which is equal to Cursor blinking rate value multiplied by 100(milliseconds).
The cursor height represents the pin height level, i.e., if you set the cursor at height level 2, that pixel will blink up to height level 2.

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses"

5.3.12 Send Image (Interruptible)

This command is used to send the actual image, such as a BMP, PNG or JPG file to the device. If the image is large, send the image in multiple packets, as in HID mode, a maximum of 4864 bytes can be sent in a single packet.

The file name must be with the file extension for example, 'map.png'. You must send vertical bar symbol (|) as a separator after the Image name.

In the image, if a data byte is the same as SOF (0x1B), send it twice to represent that byte as data and while calculating checksum do not consider this extra byte. Checksum should be sent in the last packet. Like other APIs, do not include first byte i.e., SOF in the checksum calculation.

The device gives a single short vibration at every 2 seconds after the image reception starts, until the image is displayed on the device.
In case of failure, the device will wait for 3 seconds and give an error response.

Once the image is displayed on the device, you can save it to storage media by pressing "**Space + S**" keys on the device.

First packet syntax:

Byte 0	Byte 1	Byte 2	Byte varies between 4 to 258	Separator after file name	4 Byte after file name separator (n to n+4)	Byte (n+5) onwards Up to Byte 4863
0x03	0x1B	0x2F	Name of the Image (up to 255 Bytes) including file extension	'|' (Vertical bar symbol)	Image size (4 bytes, MSB is 1st Byte and LSB is 4th Byte after separator)	Image data including header (binary form)

Intermediate packet syntax:

Byte 0	Byte 1 to Byte 4863
0x03	Image data (binary form)

Last packet syntax:

Byte 0	Byte 1 to Byte n	Byte n+1	Byte n+2	…	Byte 4862	Byte 4863
0x03	Image data (binary form) till last byte	Checksum	0x00	0x00	0x00	0x00

Response:
The device will send a success or error response as mentioned in Common Device Responses.

-

5.3.13 Send Image (Blocking)

You can use this command to send the actual image such as a BMP, PNG or JPG file to the device. If the image is large, you need to send the image in multiple packets because in HID mode, a maximum of 4864 bytes can be sent in a single packet.

Checksum should be sent in the last packet. So, while calculating checksum, you need to ignore report ID from all the packets and SOF from the first packet.

The image API is different from the other API where 0x1B checking is not required for the image data. You need to use the image data as it is, without modifying it and also provide the actual image size.

The file name must be with the file extension for example ‘map.png’. You must send vertical bar symbol (|) as a separator after Image name. The checksum calculation is the same as the other API, where you need to ignore the first two bytes i.e., Report ID and SOF. Please note that if checksum byte is 0x1B, it is necessary to send it twice.

Once the host has initiated data transmission, it must send all data bytes as per the size mentioned in “Image size” field. The device gives single short vibration at every 2 seconds after the image reception starts until the image is displayed on the device
In case of failure, the device will wait for 3 seconds and give an error response.
Once the image is displayed on the device, you can save it to storage media by pressing “**Space + s**” keys on the device.

First packet syntax:

Byte 0	Byte 1	Byte 2	Byte varies between 4 to 258	Separator after file name	4 Byte after file name separator (n to n+4)	Byte (n+5) onwards Up to Byte 4863
0x03	0x1B	0x30	Name of the Image (up to 255 Bytes) including file extension	'|' (Vertical bar symbol)	Image size (4 bytes, MSB is 1st Byte and LSB is 4th Byte after separator)	Image data including header (binary form)

Intermediate packet syntax:

Byte 0	Byte 1 to Byte 4863
0x03	Image data (binary form)

Last packet syntax:

Byte 0	Byte 1 to Byte n	Byte n+1	Byte n+2	…	Byte 4862	Byte 4863
0x03	Image data (binary form) till last byte	Checksum	0x00	0x00	0x00	0x00

Response:
The device will send a success or error response as mentioned in Common Device Responses.

5.4 *User Keys Access Commands*

5.4.1 Set Key Event

The 'Set Key Event' command is used to enable or disable the key press event. When enabled, you will get information of each key press of the device keypad. This information will include the key value and its event type.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x02	0x1B	0x31	Update Key Event 0x00 = Disable, 0x01 = Enable	0xCF: Disable 0xCE: Enable

Responses:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x31	Key Event Status after update 0x00 = Disabled, 0x01 = Enabled	0xCF: Disable 0xCE: Enable

If enabled, on each key press, you will receive the following response without any request from the host.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x07	0x1B	0x32	Key value (Higher Byte)	Key value (Lower Byte)	Key Press Type (0x02 = Short Press)	Checksum

Note: Presently Graphiti supports short key press only.

The key values sent by the device for different keys are shown in the table below:

Key	Key Value
DOT 1	0x1000
DOT 2	0x2000
DOT 3	0x4000
DOT 7	0x8000
DOT 4	0x0100
DOT 5	0x0200
DOT 6	0x0400
DOT 8	0x0800
Up	0x0010
Left	0x0020
Down	0x0040
Right	0x0080
Select	0x0001
Space	0x0002

If multiple keys are pressed together, the device will send addition of values of the pressed keys. For example, if DOT1 and DOT 4 keys are pressed together the device will send the final value as 0x1100. At the host side, bit level operation can help to decode the exact key press from the multiple key input

Key position:

DOT 3	DOT2	DOT1		Up		DOT 4	DOT 5	DOT 6
			Left	Select	Right			
				Down				
			DOT 7	Space	DOT8			

Reserved key events:

Key event	Function
DOT 5 + DOT 6 + DOT 8	Firmware upgrade
DOT 8 + Down	Hard reset
DOT 7 + DOT 8	Unregistered key event from remote device

5.5 *Touch Access Commands*

5.5.1 Set Touch Event

The 'Set Touch Event' command is used to enable or disable the touch event. When this API is enabled, you will get information of each touch event (after enabling respective mode: either gesture mode or draw mode) on the touch panel.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x02	0x1B	0x41	Set Touch Event 0x00 = Disable, 0x01 = Enable	0xBF: Disable 0xBE: Enable

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x41	Touch Event Status after update 0x00 = Disabled, 0x01 = Enabled	0xBF: Disable 0xBE: Enable

After enabling touch event and gesture mode, on each valid gesture, you will receive the following response without any request from the host.

Byte 0	Byte 1	Byte 2	Byte 4	Byte 5
0x07	0x1B	0x42	Gesture ID	Checksum

The gesture corresponding to gesture id is described in the table below:

Gesture	**Gesture ID (In decimal)**
Double Tap	2
Down Swipe	3
Up Swipe	4
Right Swipe	5
Left Swipe	6

If the draw feature is enabled along with the touch event, you will receive the following response on each touch (for drawing) without any request from the host.
Byte 3 (Length) gives the total length of data (excluding the report id, start of frames, command id and checksum byte). After the length byte is the row id, column id, pin height and blink rate of the location, where touch has been made.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	…	Byte n	Byte n+1	Byte n+2	Byte n+3	Byte n+4
0x07	0x1B	0x43	Length	Pin Row ID	Pin Column ID	Pin Height	Pin Blinking rate		Pin Row ID	Pin Column ID	Pin Height	Pin Blinking rate	Checksum

If the display is cleared, the device will send the following response.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x43	0x00	0xBD

5.5.2 Get Last Touch Point Status

The 'Get Last Touch Point Status' command is used to get the height of the pin which was last touched.
To get the last touch point status, you must first touch the pin and transmit command from the host.

The device will send response to this command only if the touch review mode is enabled.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x44	0xBC

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x07	0x1B	0x44	Pin Row ID	Pin Column ID	Pin Height	Checksum

5.6 *Vibrator Control Command*

This command is used to drive vibrators available in the device with different frequency, duty cycle and duration.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x02	0x1B	0x61	Frequency Range (10-100 kHz) (Hex)	Duty cycle in Percentage (40-100) (Hex)	Duration Higher Byte (100-1000ms) (Hex)	Duration Lower Byte (100-1000ms) (Hex)	Checksum

Response:
The device will send a success or error response as mentioned in the section "Common Device Responses".

5.7 *Clock*

5.7.1 Get Date and Time

The 'Get Date and Time' command is used to get the date and time information of the device. The device provides clock information in the 24hr format.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x02	0x1B	0x65	0xBB

Response:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
0x07	0x1B	0x65	D (ASCII)	D (ASCII)	-	M (ASCII)	M (ASCII)	-	Y (ASCII)	Y (ASCII)

Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte16	Byte 17	Byte 18
Y (ASCII)	Y (ASCII)	Space (ASCII)	H (ASCII)	H (ASCII)	:	M (ASCII)	M (ASCII)

Byte19	Byte 20	Byte 21	Byte 22
:	S (ASCII)	S (ASCII)	Checksum

5.7.2 Set Date and Time

The ‘Set Date and Time’ command is used to set the device date and time in 24-hour format. The device will accept data in 24hr format.

Syntax:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x02	0x1B	0x66	DD (Hex)	MM (Hex)	YY (Higher byte of year in hex)	YY (Lower byte of year in hex)

Byte 7	Byte 8	Byte 9	Byte 10
HH (Hex)	MM (Hex)	SS (Hex)	Checksum

Response:
The device will send a success or error response as mentioned in the section “Common Device Responses”.

6 Common Device Responses

This section details the error responses that occur under certain circumstances.

6.1 *Common responses in VCP mode*

6.1.1 Command Successful

The device will send a successful response if there is no error.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x00	0xAD

6.1.2 Command Error

The device will send a command error response in case of an invalid command byte.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x01	0xAC

Note that there is no response for improper SOF.

6.1.3 Communication Error

The device will send a communication error response as shown below.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x02	0xAB

6.1.4 Checksum Error

The device will send checksum error response if the received checksum is not matching with the calculated checksum.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x03	0xAA

6.1.5 Invalid Image API Error

The device will send an error response as shown below if the file name or file size is invalid in the image API.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x04	0xA9

6.1.6 Image API Time Out Error

The device will send a timeout error response as below in case no data is received for 3 seconds during data transmission from the host for image API.

Byte 0	Byte 1	Byte 2	Byte 3 (Checksum)
0x1B	0x53	0x05	0xA8

6.2 *Common responses in HID mode*

6.2.1 Command Successful

The device will send a successful response if there is no error.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x00	0xAD

6.2.2 Command Error

The device will send a command error response in case of an invalid command byte.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x01	0xAC

Note that there is no response for improper SOF.

6.2.3 Communication Error

The device will send a communication error response as shown below.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x02	0xAB

6.2.4 Checksum Error

The device will send a checksum error response if the received checksum is not matching the calculated checksum.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x03	0xAA

6.2.5 Invalid Image API Error

The device will send an invalid Image API error response if the file name or file size is invalid in image API.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x04	0xA9

6.2.6 Image API Time Out Error

The device will send a timeout error response in case no data is received for 3 seconds during data transmission from the host for image API.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 (Checksum)
0x07	0x1B	0x53	0x05	0xA8

7 Examples

7.1.1 Get the software version from the device

Command: 0x1B 0x01 0xFF
Response from the device: 0x1B 0x01 0x76 0x30 0x30 0x2E 0x30 0x32 0x2E 0x30 0x38 0x03

7.1.2 Blink pixel (Row = 10, Column = 10) once in every one second

Command: 0x1B 0x17 0x0A 0x0A 0x04 0x0A 0xC7
Response from the device: 0x1B 0x53 0x00 0xAD

7.1.3 Set all pixels in the 5th row at height level 4

Clear the display first
Command: 0x1B 0x16 0x03
Response from the device: 0x1B 0x53 0x00 0xAD

Set all pixels in the 5th row
Command: 0x1B 0x18 0x05 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0xF3

Response from the device: 0x1B 0x53 0x00 0xAD

7.1.4 Add one more 6th row without clearing the display

Command: 0x1B 0x18 0x06 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04
0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0x04 0x00 0xF2

Response from the device: 0x1B 0x53 0x00 0xAD

8 Limitations

- API for touch coordinates is supported in Draw mode only.

9 Appendices

9.1 *Appendix A - Revision History*

Rev.	Date	Description of Changes	Author
0.0	1st August 2017	Created	
0.1	2nd August 2017	Updated	
0.2	6th October 2017	Added limitations	
0.3	13th October 2017	Updated limitations	
0.4	14th November 2017	Updated	
0.5	25th November 2017	Modified API syntax and minor correction	
0.9	7th December 2017	Added examples	
0.10	28th February 2018	Added checksum value for the fixed messages and minor edits	
0.11	9th April 2018	Updated image API section and added error responses for image API	
0.12	10th April 2018	Edits	
0.13	18th April 2018	Edits	
0.14	1st May 2018	Modified key definitions	
0.15	28th May 2018	Modified Image API	
0.16	29th June 2018	Modified battery status API and added clock configuration API	
0.17	20th July 2018	Added Vibrator API and updated ACK/NACK section, Clock command ID, Image API section and touch API limitations	
0.18	10th August 2018	Added Interruptible Image API details	
0.19	25th September 2018	Added new VID/PID details	
0.20	30th October 2018	Minor edits	
0.21	30th November 2018	Updated API response fields	
0.22	7th July, 2021	Edits	