GTT Protocol

For all displays in the GTT Product Line

Protocol Manual

Revision 1.3

Revision History

Revision	Date	Description	Author
1.3	August 23, 2011	Added I2C	Clark
1.2	June 5, 2011	Formatting Overhaul and Minor Corrections	Clark
1.1	July 22, 2010	Revisions in wording, additional commands, correcting all return packet values and error values	JM
1.0	June 23, 2010	Initial Release	JM

Contents

1 Introduction

1.1 Communication

Serial

In serial protocol, the GTT will expect to receive eight bits of data in a package, without a parity bit, but accompanied by one stop bit. By default the unit will communicate at a baud rate of 115200bps. Finally, the display requires that hardware flow control, using RTS and CTS signals.

I^{2}C

With the I^{2}C protocol select jumpers in place, the GTT will respond to the write address specified as two hexadecimal characters in the I2C.CFG file stored within the System folder. A value incrementally higher will be assumed as the read address. The GTT can communicate as fast as 400kbps.

To write to the display, specify the write address, send one byte of data, and wait for the unit to pull the SDA line low on the ninth clock pulse to signal acknowledgement. If more data is to be sent, continue transmitting one byte at a time and waiting for the ACK.

To read, specify the read address and then read. No ACK signal will be provided so the host should be aware of how many bytes to read, or read until an unexpected zero is received. Input and output buffers are both 8,192 bytes and will discard new information when full.

Transmission

By default, all bytes sent to the display appear using the default font. For example, if the user sends character 65 to the display, an upper case ‘A’ will appear on the display. The module also responds to a few ASCII control characters while in the default mode. The display can be changed between Windows and UNIX compatibility modes using the Control Character Mode command.

Table 1: Control Characters

	UNIX Compatibility Mode	Windows Compatibility Mode
7	Trigger the Default Beep	Trigger the Default Beep
10	Move the text insertion point to the beginning of the next line down	Move the text insertion point down one line
13	Move the text insertion point to the beginning of the next line down	Move the text insertion point back to the beginning of the current line

Finally, when the display detects the command prefix character, 254, it will enter a command processing state and await the command number and its parameters. Once the command is finished, the display will automatically return to displaying all bytes sent.

Return Messages

When the display must return something to the host, it will use a standard message format. Each message will begin with the signature byte, followed by the message number, a word or double word containing the length of the message, followed by the data in the message.

The signature byte depends on the size of the data to be returned. If there is up to 65,535 bytes of data to be returned, the prefix will be 252, and the length will only be a word. If there is more data to be returned, then the prefix will be 254, and the length will be represented by a double word.

Table 2: Example Return Message

252	The return message prefix
47	Message ID 47, Get Text Colour
3	The length is sent as a word, in this case 3 bytes are expected
Byte[3]	3 bytes of data are sent representing red, green, and blue values

Error Codes

Should an error occur in the execution of a command the GTT will provide an error message to the host containing a return prefix, command number, length, and one of the error codes listed below.

Table 3: GTT Error Messages

Error Code	Description	Error Code	Description
0	Specified file was not found	1	Unrecognized bitmap file format
2	Invalid 9-slice metrics	3	Invalid 9-slice index
4	Invalid Bitmap Index	5	Invalid Bargraph Index
6	Invalid Animation Index	7	Invalid Animation file format
8	Invalid Font Index	9	Invalid Command Parameters
10	Display is out of RAM storage	11	Invalid Touch File
12	Invalid Touch Calibration	13	Successful Touch Calibration
254	Successful Command Execution	255	Unknown Exception

Touch Region Responses

Notification of touch responses is provided using a return prefix, the touch designator 135, length, one of the event types listed below, and finally the region index where the event occurred. Events not occurring within a defined region return an out of region event only. Coordinate responses are also prefixed with the return, touch, and event type bytes but provide X and Y data instead of a region index.

Table 4: GTT Touch Responses

Response	Description	Response	Description
0	Press	1	Release
2	Move	255	Out of Region

1.2 Advanced Features

Buffers

The GTT allows fonts, bargraphs, bitmaps, and animations to be easily integrated into display design. These creative touches can be developed using a variety of tools and saved to the displays SD memory card. Each font, bargraph, bitmap, or animation must then be loaded into a distinct buffer making it available for use at any time in the future. The GTT was upgraded to a buffer memory of 28MB at revision 1.1.2.3216, previously only 12MB were allocated to fonts, bargraphs, bitmaps, and animations.

Fonts

Fonts can be uploaded to the SD memory card and buffered for use on the display. The required .PE font file format can be created from regular True Type font files using Matrix Orbital’s GTT Font Builder. By default, fonts NinaB and Small are loaded on start up into buffers one and two respectively.

Bargraphs

Bargraphs simplify the display of multiple graphical representations on the screen by taking care of the calculations and placement of images. Once a bargraph is created, only the new value needs to be sent to update it. The ratio of the new value to the minimum and maximum levels is automatically calculated, and the graphic is updated. 9-Slice bargraphs also allow for nicer graphics to be used for more detailed rendering. Bargraph information is stored in a series of bargraph buffers. The index is chosen when the bargraph is created, and used to reference the bargraph in future commands.

Graphics

Bitmaps are also uploaded to the SD memory card before use. They can also be used to create touch regions or animations. Furthermore, a specific colour can be specified to appear transparent when the bitmap is rendered. Note, all file paths must be referenced with an absolute path from the root.

The 9-slice file format extends the bitmap to allow a graphic to be easily scaled. It is a simple text file that describes how to break a bitmap into nine different sections that can each be stretched or compressed to display the same image in any size required as per the representation below.

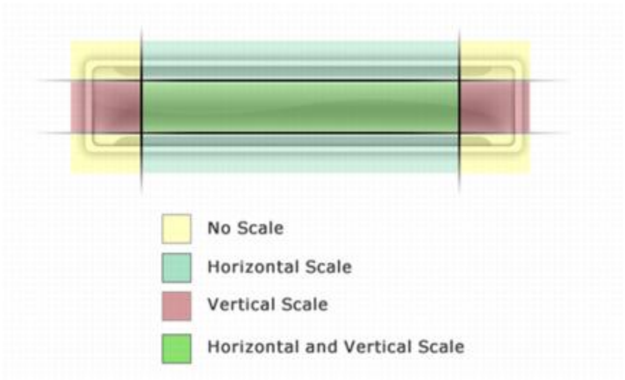

Figure 1: Adobe 9-slice Representation

Each line of a 9-slice file must start with a keyword, followed by parameters. If a line contains an unrecognized keyword, the line is ignored. The following keywords are defined:

Table 5: 9-slice Keywords

Keyword	Parameters	Description
BITMAP	1	The bitmap that will be sliced is specified
TOP	1	Specifies the lower boundary of the top slice in pixels
BOTTOM	1	Specifies the upper boundary of the bottom slice in pixels
LEFT	1	Specifies the rightmost boundary the left slice in pixels
RIGHT	1	Specifies the leftmost boundary of the right slice in pixels
VSCALE	0	If this keyword is present, the 9-slice will stretch the middle left and middle right slices to fill space as required. Without this keyword, middle will be tiled.
HSCALE	0	If this keyword is present, the 9-slice will stretch the middle top and middle bottom slices to fill space as required. Without this keyword, middle will be tiled.
TRANS-R	1	The red component of the bitmap colour to make transparent in the 9-slice
TRANS-G	1	The green component of the bitmap colour to make transparent in the 9-slice
TRANS-B	1	The blue component of the bitmap colour to make transparent in the 9-slice

Animations add yet another dimension to the graphic capability of the GTT. Like the 9-slice, a dedicated set of buffers is available for animation graphics and definition files. The animation descriptor file is also a simple text file, with a series of lines indicating frames to be displayed and their duration.

Touch

A touch screen can be configured to report events in one of two fundamental ways: region mode or coordinate mode.

Regions can be compared to buttons. A rectangular area is defined for each touch region. When that region is pressed, a report is sent back to the host and the specified “down” bitmap is displayed. After the region is released, the selected “up” bitmap is displayed.

Coordinate mode reports all events as a set of coordinates. Specific bytes are returned to designate event types such as down, move, or drag, however, no visual representation is provided.

The reporting style of both of these modes can be changed using commands found in the touch command section. Also listed are commands to save calibration data to a file, and set the current calibration from that file.

Region File

Touch regions can also be defined within a file, to allow a series of buttons to be saved and recalled as needed. Each line in the file describes a single touch region by listing the index, horizontal position, vertical position, width, height, up value, and down value, all without spaces. See the File Examples section to view a complete reference file.

1.3 The Autoexec File

On startup, the module will check the root directory of the SD card for a file named AUTOEXEC. If that file exists, it will be loaded directly into the in buffer and parsed as if it came from the input communications port.

This is useful for having custom power on defaults. Simply place the binary stream of values that the module should execute on startup in the AUTOEXEC.

If a command is started within the AUTOEXEC, however is not completed with the data in the AUTOEXEC, the display will wait for data from the serial port to complete the command. In addition, if a response is generated, flow control will wait for that byte to be handled by the host. After which, the module will return to normal operations.

1.4 Upgrades

The GTT can be quickly and easily upgraded in the field by overwriting the upgrade folder on the units SD card. Firmware upgrade files can be found at matrixorbital.ca/software, download the latest version to keep your display up to date with feature additions and bug fixes. Overwrite the upgrade folder on your GTT with the latest, cycle power to the unit, wait for the upload to complete on screen, and resume normal GTT operations.

2 Commands

2.1 Basic Commands

1.1 Change Baud Rate	Dec	254 57	Speed	v1.0.0
	Hex	FE 39	Speed	
	ASCII	■ 9	Speed	

Change the serial data rate used by the. The change is implemented immediately after the last parameter byte has been received. Baud rate will reset to 115200 on power up unless otherwise defined in the autoexec file.

Speed	Double Word	The value for the new baud rate. See table below for examples.

Table 6: Command Baud Rates

Baud Rate	Speed
9600	00 00 37 128
19200	00 00 75 00
57600	00 00 225 00
115200	00 01 194 00

1.2 Read Protocol Revision	Dec	254 00	v1.0.0
	Hex	FE 00	
	ACSII	■ NUL	

Read the version of the command protocol currently used by the display. Minor revisions will indicate an addition only while major revisions will alter or remove commands; consult the appropriate PCN for more information.

Response	Byte	Major revision of the protocol used.
	Byte	Minor revision of the protocol used.

1.3 Read Module Version	Dec	254 54	v1.0.0
	Hex	FE 36	
	ASCII	■ 6	

Read the specific firmware version currently implemented by the module.   The firmware revision of initial implementation is written next to each command in this manual.

Response	Byte	Major revision of the protocol used.
	Byte	Minor revision of the protocol used.

1.4 Reset Module	Dec	254 01	v1.0.0
	Hex	FE 01	
	ASCII	■ SOH	

Initiate a soft reset of the display. The standard start up sequence will ensue and all settings will revert to defaults.

1.5 Read Display Metrics	Dec	254 03	v1.0.0
	Hex	FE 03	
	ASCII	■ ETX	

Read the visual attributes of the display.

Response	Word	The width of the current display resolution in pixels.
	Word	The height of the current display resolution in pixels.
	Byte	The number of bits used in the red channel.
	Byte	The number of bits used in the green channel.
	Byte	The number of bits used in the blue channel.

1.6 Read Module Type	Dec	254 55	v1.0.0
	Hex	FE 37	
	ASCII	■ 7	

Return a single word that can be used to identify the display. Refer to Module Numbers below for examples.

Response	Word	The unique number of the module.

Table 7: Module Numbers

Module Type	Module Name
18,177	GTT480272A

1.7 Read Module String	Dec	254 56	v1.0.0
	Hex	FE 38	
	ASCII	■ 8	

Return a string identifying the display.

Response	String	The name of the module.

1.8 Echo	Dec	254 255	Message	v1.0.0
	Hex	FE FF	Message	
	ASCII	■	Message	

Echo a string that is sent to the display. This command can be used to test communication or indicate completion of a successful power up when placed in the autoexec file.

Message	String	An arbitrary string that the module will return.
Response	String	The same arbitrary string originally sent.

1.9 Delay	Dec	254 2	Time	v1.1.4
	Hex	FE 02	Time	
	ASCII	■ STX	Time	

Pause command execution to and responses from the display for the specified length of time.

Time	Word	Length of delay in milliseconds.

1.10 Write Customer Data	Dec	254 52	Length Data	v1.0.0
	Hex	FE 34	Length Data	
	ASCII	■ 4	Length Data	

Write up to 255 bytes of data to a file on the SD card. This data can be used for any purpose the end user sees fit including unit identification, system settings, network information, or anything else specific to the module.

Length	Byte	The length of the data to be written.
Data	Byte(s)	The data to be written to the SD Card.

1.11 Read Customer Data	Dec	254 53	v1.0.0
	Hex	FE 35	
	ASCII	■ 5	

Read data previously written using the Write Customer Data command.

Response	Byte	The length of the stored data to be read.
	Byte(s)	The data to be read from the SD Card.

1.12 Set Backlight Brightness	Dec	254 153	Brightness	v1.0.0
	Hex	FE 99	Brightness	
	ASCII	■ Ö	Brightness	

Set the brightness of the backlight.

Brightness	Byte	The backlight brightness, a value between 0 (off) and 255 (maximum).

1.13 Get Backlight Brightness	Dec	254 154	v1.0.0
	Hex	FE 9A	
	ASCII	■ Ü	

Return the current backlight brightness.

Response	Byte	The current backlight brightness.

1.14 Set Display Resolution	Dec	254 113	Width Height	v1.1.0
	Hex	FE 71	Width Height	
	ASCII	■ q	Width Height	

Set the resolution of the display.

Width	Word	Screen resolution width. Must be an even number greater than 1/8 of the native resolution.
Height	Word	Screen resolution height. Must be greater than 1/8 of the native resolution.

2.2 Drawing

2.1 Clear Screen	Dec	254 88		v1.0.0
	Hex	FE 58		
	ASCII	■ X		

Clears the screen, and resets the coordinates of both continued line and font insertion commands to zero.

2.2 Set Drawing Colour	Dec	254 99	Red Green Blue	v1.0.0
	Hex	FE 63	Red Green Blue	
	ASCII	■ c	Red Green Blue	

Set the colour to be used for all future drawing commands. The default colour on startup is white (255, 255, 255).

Red	Byte	Intensity of red, 0 to 255.
Green	Byte	Intensity of green, 0 to 255.
Blue	Byte	Intensity of blue, 0 to 255.

2.3 Get Drawing Colour	Dec	254 100	v1.0.0
	Hex	FE 64	
	ASCII	■ d	

Return the colour that is currently being used by all drawing commands.

Response	Byte	Intensity of red, 0 to 255.
	Byte	Intensity of green, 0 to 255.
	Byte	Intensity of blue, 0 to 255.

2.4 Set Background Drawing Colour	Dec	254 86	Red Green Blue	v1.0.0
	Hex	FE 56	Red Green Blue	
	ASCII	■ V	Red Green Blue	

Set the background colour to be used for all future drawing commands. The default colour on startup is black (0, 0, 0). This colour is also used as the background colour in font rendering.

Red	Byte	Intensity of red, 0 to 255.
Green	Byte	Intensity of green, 0 to 255.
Blue	Byte	Intensity of blue, 0 to 255.

2.5 Get Background Drawing Colour	Dec	254 87	v1.0.0
	Hex	FE 57	
	ASCII	■ W	

Return the background colour that is currently being used by all drawing commands.

Response	Byte	Intensity of red, 0 to 255.
	Byte	Intensity of green, 0 to 255.
	Byte	Intensity of blue, 0 to 255.

2.6 Draw Pixel	Dec	254 112	X Y	v1.0.0
	Hex	FE 70	X Y	
	ASCII	■ p	X Y	

Draw a single pixel at the specified coordinate using the current drawing colour.

X	Signed Word	Horizontal position of pixel to be drawn.
Y	Signed Word	Vertical position of pixel to be drawn.

2.7 Draw Line	Dec	254 108	X1 Y1 X2 Y2	v1.0.0
	Hex	FE 6C	X1 Y1 X2 Y2	
	ASCII	■ l	X1 Y1 X2 Y2	

Draw a line connecting two termini using the current drawing colour. Lines may be rendered differently when drawn right to left versus left to right.

X1	Signed Word	Horizontal coordinate of first terminus.
Y1	Signed Word	Vertical coordinate of first terminus.
X2	Signed Word	Horizontal coordinate of second terminus.
Y2	Signed Word	Vertical coordinate of second terminus.

2.8 Continue Line	Dec	254 101	X Y	v1.0.0
	Hex	FE 65	X Y	
	ASCII	■ e	X Y	

Draw a line from the last point drawn to the coordinate specified using the current drawing colour. The last stored point is automatically updated from Draw Point, Draw Line, and Continue Line commands.

X	Signed Word	Horizontal coordinate of terminus.
Y	Signed Word	Vertical coordinate of terminus

2.9 Draw Rectangle	Dec	254 114	X Y Width Height	v1.0.0
	Hex	FE 72	X Y Width Height	
	ASCII	■ r	X Y Width Height	

Draw a rectangular frame one pixel wide using the current drawing colour.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the rectangle.
Height	Signed Word	Height of the rectangle.

2.10 Draw Filled Rectangle	Dec	254 120	X Y Width Height	v1.0.0
	Hex	FE 78	X Y Width Height	
	ASCII	■ x	X Y Width Height	

Draw a filled rectangle using the current drawing colour.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the rectangle.
Height	Signed Word	Height of the rectangle.

2.11 Draw Circle	Dec	254 123	X Y Radius	v1.0
	Hex	FE 7B	X Y Radius	.0
	ASCII	■ {	X Y Radius	

Draw a circular frame one pixel wide using the current drawing colour.

X	Signed Word	Horizontal coordinate of circle centre.
Y	Signed Word	Vertical coordinate of circle centre.
Radius	Word	Radius of the circle.

2.12 Draw Filled Circle	Dec	254 124	X Y Radius	v1.0.0
	Hex	FE 7C	X Y Radius	
	ASCII	■ |	X Y Radius	

Draw a filled circle using the current drawing colour.

X	Signed Word	Horizontal coordinate of circle centre.
Y	Signed Word	Vertical coordinate of circle centre.
Radius	Word	Radius of the circle.

2.13 Draw Ellipse	Dec	254 125	X Y X Radius Y Radius	v1.0.0
	Hex	FE 7D	X Y X Radius Y Radius	
	ASCII	■ }	X Y X Radius Y Radius	

Draw an elliptical frame one pixel wide using the current drawing colour.

X	Signed Word	Horizontal coordinate of ellipse centre.
Y	Signed Word	Vertical coordinate of ellipse centre.
X Radius	Word	Horizontal Radius of the ellipse.
Y Radius	Word	Vertical Radius of the ellipse.

2.14 Draw Filled Ellipse	Dec	254 126	X Y X Radius Y Radius	v1.0.0
	Hex	FE 7E	X Y X Radius Y Radius	
	ASCII	■ ~	X Y X Radius Y Radius	

Draw a filled ellipse using the current drawing colour.

X	Signed Word	Horizontal coordinate of ellipse centre.
Y	Signed Word	Vertical coordinate of ellipse centre.
X Radius	Word	Horizontal Radius of the ellipse.
Y Radius	Word	Vertical Radius of the ellipse.

2.15 Draw Rounded Rectangle	Dec	254 127	X Y Width Height Radius	v1.0.0
	Hex	FE 7F	X Y Width Height Radius	
	ASCII	■ DEL		

Draw a rectangular frame one pixel wide with rounded corners using the current drawing colour. The radius must be equal to or less than half the length of the smallest side of the rectangle.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the rectangle.
Height	Signed Word	Height of the rectangle.
Radius	Word	Radius of the rounded corners.

2.16 Draw Filled Rounded Rectangle	Dec	254 128	X Y Width Height Radius	v1.0.0
	Hex	FE 80	X Y Width Height Radius	
	ASCII	■ Ç	X Y Width Height Radius	

Draw a filled rectangle with rounded corners using the current drawing colour. The radius must be equal to or less than half the length of the smallest side of the rectangle.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the rectangle.
Height	Signed Word	Height of the rectangle.
Radius	Word	Radius of the rounded corners.

2.17 Draw Triangle	Dec	254 129	Xa Ya Xb Yb Xc Yc	v1.0.0
	Hex	FE 81	Xa Ya Xb Yb Xc Yc	
	ASCII	■ ü	Xa Ya Xb Yb Xc Yc	

Draw a triangular frame one pixel wide using the current drawing colour.

Xa	Signed Word	Horizontal coordinate of the first point.
Ya	Signed Word	Vertical coordinate of the first point.
Xb	Signed Word	Horizontal coordinate of the second point.
Yb	Signed Word	Vertical coordinate of the second point.
Xc	Signed Word	Horizontal coordinate of the third point.
Yc	Signed Word	Vertical coordinate of the third point.

2.18 Enable Manual Update	Dec	254 90	Enable	v1.0.0
	Hex	FE 5A	Enable	
	ASCII	■ Z	Enable	

Stop all drawing commands from automatically updating to the screen, send them to a buffer, and allow a manual update to execute them all at one time. Useful for displaying a complicated image as a single visual update.

Enable	Byte	Any value other than 0 will enable Manual Update.

2.19 Manual Update	Dec	254 91	v1.0.0
	Hex	FE 5B	
	ASCII	■ [	

Immediately push the contents of the display buffer to the screen. This has no effect if manual update is disabled.

2.20 Region Manual Update	Dec	254 92	X Y Width Height	v1.0.0
	Hex	FE 5C	X Y Width Height	
	ASCII	■ \	X Y Width Height	

Immediately push the contents of a specified portion of the display buffer to the screen. This has no effect if manual update is disabled.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the update region.
Height	Signed Word	Height of the update region.

2.21 Scroll Screen	Dec	254 89	X Y Width Height Scroll X Scroll Y	v1.0.0
	Hex	FE 59	X Y Width Height Scroll X Scroll Y	
	ASCII	■ Y	X Y Width Height Scroll X Scroll Y	

Scroll all information in a portion of the screen.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the scroll box.
Height	Signed Word	Height of the scroll box.
Scroll X	Signed Word	Horizontal distance to scroll.
Scroll Y	Signed Word	Vertical distance to scroll.

2.3 Buffers

3.1 Load Bitmap	Dec	**254 95**	**Index Filename**	**v1.0.0**
	Hex	**FE 5F**	**Index Filename**	
	ASCII	**■ _**	**Index Filename**	

Load a bitmap into the specified buffer. Supported formats are GIF, BMP, and JPG (in RGB format only).

Index	**Byte**	Buffer index where the bitmap is loaded.
Filename	**String**	Filename, and path from the root folder, of the bitmap file to load.

3.2 Load 9-Slice	Dec	**254 144**	**Index Filename**	**v1.0.0**
	Hex	**FE 90**	**Index Filename**	
	ASCII	**■ É**	**Index Filename**	

Load a 9-Slice file into the specified buffer. Refer to the 9-slice file example for more info.

Index	**Byte**	Buffer index where the 9-Slice is loaded.
Filename	**String**	Filename, and path from the root folder, of the 9-Slice file to load.

3.3 Load Font	Dec	**254 40**	**Index Filename**	**v1.0.0**
	Hex	**FE 28**	**Index Filename**	
	ASCII	**■ (**	**Index Filename**	

Load a font file into the buffer. Fonts must be converted into .pe format using the GTT Support Tool.

Index	**Byte**	Buffer index where the font is loaded.
Filename	**String**	Filename, including path from the root folder, of the font file to load.

3.4 Load Animation	Dec	**254 192**	**Index Filename**	**v1.0.0**
	Hex	**FE C0**	**Index Filename**	
	ASCII	**■ └**	**Index Filename**	

Load an animation file into the buffer. Refer to **Error! Reference source not found.** for more information.

Index	**Byte**	Buffer index where the animation is loaded.
Filename	**String**	Filename, and path from the root folder, of the animation file to load.

3.5 Copy Screen Rectangle	Dec	**254 96**	**Index X Y Width Height**	**v1.0.0**
	Hex	**FE 60**	**Index X Y Width Height**	
	ASCII	**■ `**	**Index X Y Width Height**	

Save a rectangular region of the screen to the specified bitmap buffer, allowing it to be recalled at a later time.

Index	**Byte**	Bitmap buffer index where the screen section is saved.
X	**Signed Word**	Leftmost coordinate.
Y	**Signed Word**	Topmost coordinate.
Width	**Signed Word**	Width of the screen section.
Height	**Signed Word**	Height of the screen section.

3.6 Clear A Buffer

Dec	254 208	Type Index	v1.0.0
Hex	FE D0	Type Index	
ASCII	■ ╨	Type Index	

Clear all information from a selected buffer index, freeing RAM.

Type	Byte	Type of buffer to clear, as per the table below.
Index	Byte	Buffer index to be cleared.

Table 8: Buffer Types

Type	Description
0	Animations
1	Bitmaps
2	9-Slices
3	Fonts

3.7 Clear All Buffers

Dec	254 209	v1.0.0
Hex	FE D1	
ASCII	■ ╤	

Clear all information from all buffers, freeing all RAM.

2.4 Fonts

4.1 Set Font	Dec	254 49	Buffer	v1.0.0
	Hex	FE 31	Buffer	
	ASCII	■ 1	Buffer	

Set the current font to one that has been loaded into the buffer specified.

Buffer	Byte	Buffer index where the desired font file is loaded.

4.2 Get Font	Dec	254 48	v1.0.0
	Hex	FE 30	
	ASCII	■ 0	

Get the buffer index of the current font.

Response	Byte	Buffer index where the current font file is loaded.

4.3 Set Font Metrics	Dec	254 50	Vertical Pitch	Horizontal Pitch	v1.0.0
	Hex	FE 32	Vertical Pitch	Horizontal Pitch	
	ASCII	■ 2	Vertical Pitch	Horizontal Pitch	

Set the font metrics, or spacing. Positive pitch will add space between characters, negative pitch will subtract.

Vertical Pitch	Signed Byte	Number of pixels between rows of text.
Horizontal Pitch	Signed Byte	Number of pixels between adjacent characters.

4.4 Get Font Metrics	Dec	254 41	v1.0.0
	Hex	FE 29	
	ASCII	■)	

Return the font metrics, or spacing values, currently in use.

Response	Signed Byte	Number of pixels between rows of text.
	Signed Byte	Number of pixels between adjacent characters.

4.5 Set Text Colour	Dec	254 46	Red	Green	Blue	v1.0.0
	Hex	FE 2E	Red	Green	Blue	
	ASCII	■ .	Red	Green	Blue	

Set the colour of all subsequent text, current text is not affected.

Red	Byte	Intensity of red, 0 to 255.
Green	Byte	Intensity of green, 0 to 255.
Blue	Byte	Intensity of blue, 0 to 255.

4.6 Get Text Colour	Dec	254 47		v1.0.0
	Hex	FE 2F		
	ASCII	■ /		
Return the current text colour in use.				
Response	Byte	Intensity of red, 0 to 255.		
	Byte	Intensity of green, 0 to 255.		
	Byte	Intensity of blue, 0 to 255.		

4.7 Calculate String Extents	Dec	254 42	String	v1.0.0
	Hex	FE 2A	String	
	ASCII	■ *	String	
Return the size of the rectangle that the specified string would occupy if it was rendered with the current font. This is useful for positioning and clearing text on the display.				
String	String	String whose extents are desired.		
Response	Word	Width of the rendered string.		
	Word	Height of the rendered string.		

4.8 Set Text Window	Dec	254 43	X Y Width Height	v1.0.0
	Hex	FE 2B	X Y Width Height	
	ASCII	■ +	X Y Width Height	
Create an invisible rectangle to which future text will be confined. Text will follow wrap and scroll settings.				
X	Signed Word	Leftmost coordinate.		
Y	Signed Word	Topmost coordinate.		
Width	Signed Word	Width of the text window.		
Height	Signed Word	Height of the text window.		

4.9 Get Text Window	Dec	254 44		v1.0.0
	Hex	FE 2C		
	ASCII	■ ,		
Return the extents of the current text window.				
Response	Signed Word	Leftmost coordinate.		
	Signed Word	Topmost coordinate.		
	Signed Word	Width of the text window.		
	Signed Word	Height of the text window.		

4.10 Set Text Insertion Point	Dec	254 121	X Y	v1.0.0
	Hex	FE 79	X Y	
	ASCII	■ y	X Y	

Define the upper left coordinate of the next text string to be displayed, clipped to the current text window.

X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.

4.11 Get Text Insertion Point	Dec	254 122	v1.0.0
	Hex	FE 7A	
	ASCII	■ z	

Return the upper left coordinate of the next text string to be displayed.

Response	Signed Word	Leftmost coordinate.
	Signed Word	Topmost coordinate.

4.12 Set Scroll Mode	Dec	254 81	Mode	v1.0.0
	Hex	FE 51	Mode	
	ASCII	■ Q	Mode	

Define how text will behave when it reaches the bottom of the text window. Invalid modes will default to 2.

Mode	Byte	Scrolling and wrapping settings; see Text Behaviour below.

Table 9: Text Behaviour

Mode	Description
0	No scrolling, additional text will appear at the bottom of the window.
1	No scrolling, additional text will appear at the top of the window.
2	Scroll up to make room for additional text at the bottom of the window.

4.13 Get Scroll Mode	Dec	254 82	v1.0.0
	Hex	FE 52	
	ASCII	■ R	

Return the scroll mode currently in use.

Response	Byte	Scrolling and wrapping settings; see Text Behaviour above.

4.14 Go Home	Dec	254 72	v1.0.0
	Hex	FE 48	
	ASCII	■ H	

Set the text insertion point to the upper leftmost corner of the current text window.

4.15 Set Control Character Mode	Dec	254 38	Mode	v1.0.0
	Hex	FE 26	Mode	
	ASCII	■ &	Mode	

Set the behavior of defined control characters. Default is Unix, mode 0.

Mode	Byte	0 for Unix mode or 1 for Windows mode; see Control Character Modes below.

Table 10: Control Character Modes

	UNIX Compatibility Mode	Windows Compatibility Mode
7	The bell character will signal the Default Beep	The bell character will signal the Default Beep
10	Move the text insertion point to the beginning of the next line down	Move the text insertion point down one line
13	Move the text insertion point to the beginning of the next line down	Move the text insertion point back to the beginning of the current line

4.16 Get Control Character Mode	Dec	254 39	v1.0.0
	Hex	FE 27	
	ASCII	■ ‘	

Return the current control character mode.

Response	Byte	Current control character mode; see Control Character Modes above.

4.17 Reset Font	Dec	254 45	v1.0.0
	Hex	FE 2D	
	ASCII	■ -	

Restores default fonts NinaB and Small to indices 1 and 2 respectively.

2.5 Graphics

5.1 Display Bitmap	Dec	254 97	Index X Y	v1.0.0
	Hex	FE 61	Index X Y	
	ASCII	■ a	Index X Y	

Displays a bitmap previously loaded into the specified buffer index.

Index	Byte	Buffer index where the bitmap is loaded.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.

5.2 Display 9-Slice	Dec	254 145	Index X Y Width Height	v1.0.0
	Hex	FE 91	Index X Y Width Height	
	ASCII	■ æ	Index X Y Width Height	

Displays a 9-slice previously loaded into the specified buffer index.

Index	Byte	Buffer index where the 9-slice is loaded.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the 9-slice.
Height	Signed Word	Height of the 9-slice.

5.3 Set Bitmap Transparency	Dec	254 98	Index Red Green Blue	v1.0.0
	Hex	FE 62	Index Red Green Blue	
	ASCII	■ b	Index Red Green Blue	

Set a specified colour of one identified bitmap to be transparent.

Index	Byte	Buffer index where the bitmap is loaded.
Red	Byte	Intensity of red, 0 to 255.
Green	Byte	Intensity of green, 0 to 255.
Blue	Byte	Intensity of blue, 0 to 255.

5.4 Display Animation	Dec	254 193	Index Animation X Y	v1.0.0
	Hex	FE C1	Index Animation X Y	
	ASCII	■ ┴	Index Animation X Y	

Load the first frame of the specified animation in its stopped state at the specified location. If an animation is already in use at that index it will be overwritten. Use the start animation command to play the displayed file.

Index	Byte	Buffer index where the animation file is loaded.
Animation	Byte	Animation index where state information will be stored.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.

5.5 Start/Stop Animation

Dec	254 194	Animation Start	v1.0.0
Hex	FE C2	Animation Start	
ASCII	■ ┬	Animation Start	

Start or stop an animation that has been displayed.

Animation	Byte	Animation index to be started/stopped.
Start	Byte	Any non-zero value will start the specified animation, 0 will stop it.

5.6 Set Animation Frame

Dec	254 195	Animation Frame	v1.0.0
Hex	FE C3	Animation Frame	
ASCII	■ ├	Animation Frame	

Set the current frame of a displayed animation. If the frame exceeds the total number present, the animation will be set to the first frame.

Animation	Byte	Animation index upon which the current frame will be changed.
Frame	Byte	Number of the frame to be displayed.

5.7 Get Animation Frame

Dec	254 196	Animation	v1.0.0
Hex	FE C4	Animation	
ASCII	■ ─	Animation	

Returns the current frame of the specified animation.

Animation	Byte	Animation index upon which the current frame query will be carried out.
Response	Byte	Current animation frame.

5.8 Stop All Animations

Dec	254 198	v1.0.0
Hex	FE C6	
ASCII	■ ╞	

Stop all currently running animations at their present frame.

5.9 Delete Animation

Dec	254 199	Animation	v1.0.0
Hex	FE C7	Animation	
ASCII	■ ╟	Animation	

Stop and delete the displayed animation specified.

Animation	Byte	Animation index to be deleted.

5.10 Delete All Animations

Dec	254 200	v1.0.0
Hex	FE C8	
ASCII	■ ╚	

Stop and delete all displayed animations. Animations loaded into buffers will remain, unless buffers are cleared.

5.11 Define A Plain Bargraph	Dec	254 103	Index Min Max X Y Width Height FG FR FB BR BG BB Dir	v1.0.0
	Hex	FE 67	Index Min Max X Y Width Height FG FR FB BR BG BB Dir	
	ASCII	■ g	Index Min Max X Y Width Height FG FR FB BR BG BB Dir	

Define a new plain bargraph indentified by the specified index. New indices will overwrite old, invalid directions will default to 0, and inverted min and max values will be automatically corrected.

Index	Byte	Buffer index where the bitmap is to be loaded.
Min	Signed Word	Minimum bargraph value.
Max	Signed Word	Maximum bargraph value.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the bargraph.
Height	Signed Word	Height of the bargraph.
FR	Byte	Red component of the foreground colour.
FG	Byte	Green component of the foreground colour.
FB	Byte	Blue component of the foreground colour.
BR	Byte	Red component of the background colour.
BG	Byte	Green component of the background colour.
BB	Byte	Blue component of the background colour.
Dir	Byte	Direction that the bargraph will take, see Bargraph Orientation below.

Table 11: Bargraph Orientation

Direction	Description
0	Bottom to Top
1	Left to Right
2	Right to Left
3	Top to Bottom

5.12 Define A 9-Slice Bargraph	Dec	254 104	Index Min Max X Y Width Height FBuffer BBuffer Dir	v1.0.0
	Hex	FE 68	Index Min Max X Y Width Height FBuffer BBuffer Dir	
	ASCII	■ h	Index Min Max X Y Width Height FBuffer BBuffer Dir	

Define a new 9-slice bargraph indentified by the specified index. New indices will overwrite old, invalid directions will default to 0, and inverted min and max values will be automatically corrected.

Index	Byte	Buffer index where the bitmap is to be loaded.
Min	Signed Word	Minimum bargraph value.
Max	Signed Word	Maximum bargraph value.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the bargraph.
Height	Signed Word	Height of the bargraph.
FBufer	Byte	Buffer index of the foreground image.
BBuffer	Byte	Buffer index of the background image.
Dir	Byte	Direction that the bargraph will take, see Bargraph Orientation above.

5.13 Update A Bargraph Value

Dec	254 105	Index Value	v1.0.0
Hex	FE 69	Index Value	
ASCII	■ i	Index Value	

Update the value of the specified bargraph. Value will be bounded to the bargraph minimum and maximum.

Index	Byte	Index of the bargraph to update.
Value	Word	Value of the bargraph.

5.14 Update Multiple Bargraph Values

Dec	254 106	Index Count Values	v1.0.0
Hex	FE 6A	Index Count Values	
ASCII	■ j	Index Count Values	

Update the value of a number of bargraphs beginning at the specified index. Value will be bounded to the bargraph minimum and maximum.

Index	Byte	Index of the first bargraph to update.
Count	Byte	Number of successive bargraphs to update.
Value	Words	A specific value for each bargraph.

5.15 List All Bargraphs

Dec	254 102	v1.0.0
Hex	FE 66	
ASCII	■ f	

Return a complete list of all bargraphs saved to buffers, three bytes per entry indicate type, use, and current value.

Response	Byte	Type of bargraph used as per Bargraph Types below.
	Word	Current graph value, 0 is returned if graph is unused.

Table 12: Bargraph Types

Type	Description
0	Unused
1	Plain
2	9-slice

5.16 Clear All Bargprahs

Dec	254 107	v1.0.0
Hex	FE 6B	
ASCII	■ k	

Clear all bargraphs from buffers, setting all to the unused state and erasing all attributes from memory.

2.6 Touch

6.1 Calibrate Touch Screen	Dec	254 139		v1.0.0
	Hex	FE 8B		
	ASCII	■ ï		

Initiate the touch screen calibration sequence, after user input is complete a confirmation byte will be returned and the calibration will be saved as “\SYSTEM\touchcal.dat”. Calibration can be restored from the file at any time.

Response	Byte	A value of 1 is returned if calibration is successful.

6.2 Restore Touch Calibration	Dec	254 141		v1.0.0
	Hex	FE 8D		
	ASCII	■ ì		

Restore touch calibration using the data from “\SYSTEM\touchcal.dat”, if this file is present.

Response	Byte	A value of 1 is returned if the file is found and calibration is successfully restored.

6.3 Create A Touch Region	Dec	254 132	Index X Y Width Height Up Down	v1.0.0
	Hex	FE 84	Index X Y Width Height Up Down	
	ASCII	■ ä	Index X Y Width Height Up Down	

Create a rectangular region on the screen that will report a unique index for touch events occurring within its bounds. While the region is inactive the up image will be displayed, when it is pressed the down will appear.

Index	Byte	Buffer index where the touch region is loaded.
X	Signed Word	Leftmost coordinate.
Y	Signed Word	Topmost coordinate.
Width	Signed Word	Width of the touch region.
Height	Signed Word	Width of the touch region.
Up	Byte	Index of the loaded bitmap displayed when the region is untouched.
Down	Byte	Index of the loaded bitmap displayed when the region is touched.

6.4 Clear A Touch Region	Dec	254 133	Index	v1.0.0
	Hex	FE 85	Index	
	ASCII	■ à	Index	

Clear the specified touch region from the screen and memory, ensuring its touch events will no longer be reported.

Index	Byte	Buffer index where the touch region is loaded.

6.5 Clear All Touch Regions	Dec	254 134		v1.0.0
	Hex	FE 86		
	ASCII	■ å		

Clear all touch regions from the screen and memory, ensuring their touch events will no longer be reported.

6.6 Load Region File	Dec	254 140	Filename	v1.0.0
	Hex	FE 8C	Filename	
	ASCII	■ î	Filename	

Load a group of touch region definitions from a file, old regions are cleared. See File Examples for more info.

Filename	String	Filename, and path from the root folder, of the region file to load.

6.7 Change Touch Reporting Style	Dec	254 135	Style	v1.0.0
	Hex	FE 87	Style	
	ASCII	■ ç	Style	

Customize the way in which touch events are reported.

Style	Byte	Select one of the reporting styles from Touch Reporting Styles below .

Table 13: Touch Reporting Styles

Style	Description
0	No event reporting
1	Down region events only
2	Up region events only
3	Down and up region events
9	Down coordinate events only
10	Up coordinate events only
11	Down and up coordinate events

6.8 Set Out of Region Setting	Dec	254 142	Setting	v1.1.1
	Hex	FE 8E	Setting	
	ASCII	■ Ä	Setting	

Set whether out of region responses will be returned or not.

Setting	Byte	Zero will disable out of region responses while any non-zero value will enable them.

6.9 Get Out of Region Setting	Dec	254 143	v1.1.1
	Hex	FE 8F	
	ASCII	■ Å	

Customize the way in which touch events are reported.

Response	Byte	Current out of region setting, zero will indicate out of region responses are disabled while any non-zero value will confirm them to be active.

2.7 Sound

7.1 Buzzer Beep	Dec	254 187	Frequency Duration	v1.0.0
	Hex	FE BB	Frequency Duration	
	ASCII	■ ╗	Frequency Duration	
Generate a tone from the piezo buzzer at the specified frequency for the determined interval.				
Frequency	Word	Frequency of the beep in Hertz.		
Duration	Word	Duration of the beep in milliseconds.		

7.2 Set Default Buzzer Beep	Dec	254 188	Frequency Duration	v1.0.0
	Hex	FE BC	Frequency Duration	
	ASCII	■ ╝	Frequency Duration	
Set the frequency and duration of the default beep transmitted when the bell character is transmitted.				
Frequency	Word	Frequency of the beep in Hertz.		
Duration	Word	Duration of the beep in milliseconds.		

2.8 Filesystem

8.1 Delete a File	Dec	254 240	Filename	v1.0.2
	Hex	FE F0	Filename	
	ASCII	■ ≡	Filename	
Delete the specified file from the SD card, the byte 254 will be returned upon success. Use with caution.				
Filename	String	The name of the file to delete.		

8.2 Rename a File	Dec	254 241	Source Destination	v1.0.2
	Hex	FE F1	Source Destination	
	ASCII	■ ±	Source Destination	
Rename or move the original source file to the destination.				
Source	String	The original filename to rename or move.		
Destination	String	The new filename to move the source to.		

8.3 Create a Folder	Dec	254 242	Filename	v1.0.2
	Hex	FE F2	Filename	
	ASCII	■ ≥	Filename	
Create a folder on the SD card with the specified name.				
Filename	String	The name of the folder to create.		

8.4 Download a File	Dec	254 243	Filename	v1.0.2
	Hex	FE F3	Filename	
	ASCII	■≤	Filename	

Download a file from the SD card.

Filename	String	Name of the file to download.
Response	Double Word	Length in bytes of the file.
	Byte(s)	Contents of the file transmitted via Xmodem protocol.

8.5 Upload a File	Dec	254 244	Length Filename Data	v1.0.2
	Hex	FE F4	Length Filename Data	
	ASCII	■⌠	Length Filename Data	

Upload a file to the SD card.

Length	Word	Name of the file to upload.
Filename	Word	Length in bytes of the file.
Data	Byte(s)	Contents of the file transmitted via Xmodem protocol.

8.6 List File Directory	Dec	254 245	Path	v1.0.2
	Hex	FE F5	Path	
	ASCII	■⌡	Path	

List the files, their sizes, and subfolders within the specified path. One response is sent for each file entry.

Path	String	The path for which directories will be listed.
Response	Word	The length of the entire return packet.
	Word	*The number of directories to expect.
	String	The name of the directory.
	Word	The number of files to expect.
	String	The filename.
	Double Word	The size of the file in bytes.

***Note:** If the second word is 0, the following string is omitted completely. If the third word is 0, then the following string and double word are omitted completely.

3 Appendix

3.1 Command Summary

Available commands below include identifying number, required parameters, the returned response and the response type.

Table 14: Basic Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Change Baud Rate	57	39	9	Double Word	None
Read Protocol Revision	00	00	NUL	None	Byte [2]
Read Module Version	54	36	6	None	Byte [2]
Reset Module	1	01	SOH	None	None
Read Display Metrics	3	03	ETX	None	Word [2], Byte [3]
Read Module Type	55	37	7	None	Word
Read Module String	56	38	8	None	String
Echo	255	FF		String	String
Delay	2	02	STX	Word	None
Write Customer Data	52	34	4	Byte []	None
Read Customer Data	53	35	5	None	Byte []
Set Backlight Brightness	153	99	Ö	Byte	None
Get Backlight Brightness	154	9A	Ü	None	Byte
Set Display Resolution	113	71	q	Word [2]	None

Table 15: Drawing Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Clear Screen	88	58	X	None	None
Set Drawing Colour	99	63	c	Byte [3]	None
Get Drawing Colour	100	64	d	None	Byte [3]
Set Background Drawing Colour	86	56	V	Byte [3]	None
Get Background Drawing Colour	87	57	W	None	Byte [3]
Draw Pixel	112	70	p	Signed Word [2]	None
Draw Line	108	6C	l	Signed Word [4]	None
Continue Line	101	65	e	Signed Word [2]	None
Draw Rectangle	114	72	r	Signed Word [4]	None
Draw Filled Rectangle	120	78	x	Signed Word [4]	None
Draw Circle	123	7B	{	Signed Word [2], Word	None
Draw Filled Circle	124	7C	|	Signed Word [2], Word	None
Draw Ellipse	125	7D	}	Signed Word [2], Word [2]	None
Draw Filled Ellipse	126	7E	~	Signed Word [2], Word [2]	None
Draw Rounded Rectangle	127	7F	DEL	Signed Word [4], Word	None
Draw Filled Rounded Rectangle	128	80	Ç	Signed Word [4], Word	None
Draw Triangle	129	81	ü	Signed Word [6]	None
Enable Manual Update	90	5A	Z	Byte	None
Manual Update	91	5B	[	None	None
Region Manual Update	92	5C	\	Signed Word [4]	None
Scroll Screen	89	59	Y	Signed Word [6]	None

Table 16: Buffers Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Load Bitmap	95	5F	_	Byte, String	None
Load 9-Slice	144	90	É	Byte, String	None
Load Font	40	28	(	Byte, String	None
Load Animation	192	C0	└	Byte, String	None
Copy Screen Rectangle	96	60	`	Byte, Signed Word [4]	None
Clear A Buffer	208	D0	╨	Byte [2]	None
Clear All Buffers	209	D1	╤	None	None

Table 17: Text Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Set Font	49	31	1	Byte	None
Get Font	48	30	0	None	Byte
Set Font Metrics	50	32	2	Signed Byte [2]	None
Get Font Metrics	41	29	)	None	Signed Byte [2]
Set Text Colour	46	2E	.	Byte [3]	None
Get Text Colour	47	2F	/	None	Byte [3]
Calculate String Extents	42	2A	*	String	Word [2]
Set Text Window	43	2B	+	Signed Word [4]	None
Get Text Window	44	2C	,	None	Signed Word [4]
Set Text Insertion Point	121	79	y	Signed Word [2]	None
Get Text Insertion Point	122	7A	z	None	Signed Word [2]
Set Scroll Mode	81	51	Q	Byte	None
Get Scroll Mode	82	52	R	None	Byte
Go Home	72	48	H	None	None
Set Control Character Mode	38	26	&	Byte	None
Get Control Character Mode	39	27	‘	None	Byte
Reset Font	45	2D	-	None	None

Table 18: Graphics Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Display Bitmap	97	61	a	Byte, Signed Word [2]	None
Display 9-Slice	145	91	æ	Byte, Signed Word [4]	None
Set Bitmap Transparency	98	62	b	Byte [4]	None
Display Animation	193	C1	┴	Byte [2], Signed Word [2]	None
Start/Stop Animation	194	C2	┬	Byte [2]	None
Set Animation Frame	195	C3	├	Byte [2]	None
Get Animation Frame	196	C4	─	Byte	Byte
Stop All Animations	198	C6	╞	None	None
Delete Animation	199	C7	╟	Byte	None
Delete All Animations	200	C8	╚	None	None
Define A Plain Bargraph	103	67	g	Byte, Signed Word [6], Byte [7]	None
Define A 9-Slice Bargraph	104	68	h	Byte, Signed Word [6], Byte [3]	None
Update A Bargraph Value	105	69	i	Byte, Word	None
Update Multiple Bargraph Values	106	6A	j	Byte [2], Word []	None
List All Bargraphs	102	66	f	None	Byte [] [3]
Clear All Bargprahs	107	68	k	None	None

Table 19: Touch Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Calibrate Touch Screen	139	8B	ï	None	Byte
Restore Touch Calibration	141	8D	Ì	None	Byte
Create A Touch Region	132	84	ä	Byte, Signed Word [4], Byte [2]	None
Clear A Touch Region	133	85	à	Byte	None
Clear All Touch Regions	134	86	å	None	None
Load Region File	140	8C	î	String	None
Change Touch Reporting Style	135	87	ç	Byte	None
Set Out of Region Setting	142	8E	Ä	Byte	None
Get Out of Region Setting	143	8F	Å	None	Byte

Table 20: Sound Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Buzzer Beep	187	BB	╗	Word [2]	None
Set Default Buzzer Beep	188	BC	╝	Word [2]	None

Table 21: Filesystem Command Summary

Name	Dec	Hex	ASCII	Parameters	Response
Delete a File	240	F0	≡	String	None
Rename a File	241	F1	±	String [2]	None
Create a Folder	242	F2	≥	String	None
Download a File	243	F3	≤	String	None
Upload a File	244	F4	⌠	Word [2], Byte	None
List File Directory	245	F5	⌡	String	See List File Directory

3.2 Data Types

Common Language Representations

The following table outlines native data types in common programming languages that can be used to represent the data types used in this manual.

Table 22: Data Types with Representations

	ANSI C/C++	C#	Visual Basic
Byte	unsigned char	byte	Byte
Word	unsigned short int	Unsigned short	UShort
Signed Word	short int	Short	Short
Double Word	unsigned int	Unsigned int	UInteger
Signed Double Word	int	Int	Integer
String	string	String	String

Table 23: Data Type Descriptions

Byte	Unsigned 8 bit data type that can represent value from 0 to 255.
Word	Unsigned two byte type can represent values from 0 to 65,536. The bytes are always communicated in big endian order.
Signed Word	Two bytes that can represent values from -32,768 to 32,767. The values are stored in two's complement form.
Double Word	Four bytes, and can represent values from 0 to 4,294,967,295.
Signed Double Word	Four bytes, and can be represent values of -2,147,483,648 to 2,147,483.
String	Strings are a length of bytes terminated by a single null byte. The ASCII character set is used..

3.3 Colour Codes

Table 24: Basic Colour Codes

Colour	Red	Green	Blue	Colour	Red	Green	Blue
Red	255	0	0	White	255	255	255
Turquoise	0	255	255	Light Grey	192	192	192
Light Blue	0	0	255	Dark Grey	128	128	128
Dark Blue	0	0	160	Black	0	0	0
Light Purple	255	0	128	Orange	255	128	64
Dark Purple	128	0	128	Brown	128	64	0
Yellow	255	255	0	Burgundy	128	0	0
Pastel Green	0	255	0	Forest Green	128	128	0
Pink	255	0	255	Grass Green	64	128	128

3.4 File Examples

9-slice

The following file creates a 9-slice map for the button.bmp image located in the Bitmap folder of the root SD card director.

```
BITMAP \Bitmaps\button.bmp
TOP 2
BOTTOM 2
LEFT 4
Right 4
```

This file specifies that the top and bottom regions be two pixels in height while the left and right regions are each four pixels in width. The center of the image will be tiled and no colour will be transparent.

Animation

This animation is comprised of four bitmaps each saved in an ani1 folder within an Animations folder.

```
10 \Animations\ani1\frame1.bmp
20 \Animations\ani1\frame2.bmp
10 \Animations\ani1\frame3.bmp
100 \Animations\ani1\frame4.bmp
```

The above example would define a simple animation with 6 frames, each frame displayed for 10ms, except for frames 2 and 4 which are displayed for 20ms and 100ms respectively.

Region

This would define two touch regions.

```
1 10 10 40 10 1 2
2 10 30 50 20 3 4
```

Region 1 is positioned at coordinates 10, 10, with a width of 40 pixels and a height of 10. When it is pressed the bitmap in bitmap buffer 2 will be displayed, and when it is not pressed bitmap 1 will be displayed. Region 2 is at coordinates 10, 30 with a width of 50 pixels and a height of 20. It uses bitmaps 3 and 4 for it's up and down indicators respectively

Autoexec

Autoexec files are created as a stream of bytes appearing exactly as they would if they were sent to the display over the serial port. The following is an autoexec file as viewed using a hex editor.

```
FE BB 03 E8 01 F4 48 65 6C 6C 6F 20 57 6F 72 6C 64 21
```

This file executes a buzzer beep at a frequency of 1000Hz for 500ms before transmitting a "Hello World!" string. It may be difficult to view unprintable characters such as the command prefix using a simple text editor such as notepad so a hex editor is recommended for viewing files and creating scripts if the GTT Support tool is not used.

4 Definitions

ASCII: American standard code for information interchange used to give standardized numeric codes to alphanumeric characters.

Big Endian: Transmission protocol whereby the most significant byte is transmitted first.

BPS: Bits per second, a measure of transmission speed.

I^2C: Inter-integrated circuit protocol uses clock and data lines to communicate short distances at slow speeds from a master to up to 128 addressable slave devices. A display is a slave device.

LSB: Least significant bit or byte in a transmission, the rightmost when read.

MSB: Most significant bit or byte in a transmission, the leftmost when read.

RS232: Recommended standard 232, a common serial protocol. A low level is -30V, a high is +30V.

RS422: Recommended standard 422, a more robust differential pair serial protocol.

SDA: Serial data line used to transfer data in I^2C protocol. This open drain line should be pulled high through a resistor. Nominal values are between 1K and 10K Ω.

SCL: Serial clock line used to designate data bits in I^2C protocol. This open drain line should be pulled high through a resistor. Nominal values are between 1K and 10K Ω.

TTL: Transistor-transistor logic applied to serial protocol. Low level is 0V while high logic is 5V.

TFT: Thin film transistor with reference to an LCD technology.

5 Contact

Sales

Phone: 403.229.2737
Email: sales@matrixorbital.ca

Design

Phone: 403.229.2737
Email: design@matrixorbital.ca

Support

Phone: 403.229.2737
Email: support@matrixorbital.ca

Online

Purchasing: www.matrixorbital.com
Support: www.matrixorbital.ca