MATRIX ORBITAL

# GTT Protocol

For all variants of the GTT35A, GTT38A, GTT43A, GTT50A & GTT70A

## Protocol Manual

Revision 2.2

**Firmware Revision: 2.0 or Higher**

# Revision History

| Revision | Date | Description | Author |
|---|---|---|---|
| 2.2 | 21 October 2014 | Added Region Activation Features, Improved Readability | Clark |
| 2.1 | 8 April 2014 | Added Scripting, Label, and Trace Features | Clark |
| 2.0 | 8 October 2013 | Initial Release | Clark |

# Contents

# 1 Introduction

## 1.1 Communication

### Settings

The multiple communication protocols available and simple command structure of the GTT mean that a variety of applications can be used to communicate with the display. Basic default settings for serial protocols, including USB, TTL, RS232, and RS422, as well as I$^2$C are shown below.

*Table 1: Serial Communication Settings*

| Speed | Data Bits | Parity | Stop Bits | Flow Control |
|---|---|---|---|---|
| 115.2Kbps | 8 | None | 1 | Hardware |

*Table 2: I$^2$C Communication Settings*

| Write | Read | Speed |
|---|---|---|
| $80_d$ | $81_d$ | Up to 400KHz |

### Transmission

By default, all bytes sent to the display are printed using the default font and standard ASCII encoding. For example, if the user sends the decimal value 65 to the display, an upper case ‘A’ will appear on the display. The module also responds to a few ASCII control characters. The functionality of these characters can be toggled between Windows and UNIX compatibility modes using the Control Character Mode command. The default behavior is Unix mode.

*Table 3: Control Characters*

| | 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 |

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. Multiple bytes are transferred in Big Endian format. Once the command is finished, the display will automatically return to displaying all bytes sent.

### Return Messages

When the display must return data to the host, it will use a standard message format. Each message will begin with the return message prefix 252, followed by the command number generating the message, a short value containing the length of the data in the message, and finally the data in the message.

*Table 4: Example Return Message*

| Prefix | Message ID | Length | | Data | | |
|---|---|---|---|---|---|---|
| 252 | 47 | 0 | 3 | 255 | 255 | 255 |

The sample above shows the expected return values from the Get Font Colour command. In this manual, expected return messages are described below any required parameters.

## 1.2 Basic Features

### Drawing

The most basic commands available for the GTT line are the drawing features. Simple shapes, from pixels to triangles, can easily be drawn on the unit using a number of available commands. It should be noted that the coordinate system of the GTT references the top left pixel as 0,0 and increments positively to the right and down, as shown below.

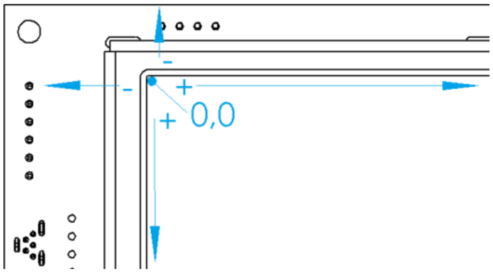


*Figure 1: Pixel coordinate orientation (GTT43A shown)*

The drawing colour can be set globally by specifying values for red, green, and blue channels, and will default to white. The Get Display Metrics function will report the number of bits available for each colour channel. The GTT will use the highest bits of any colour specified, dropping the lowest if necessary. For example, if the display uses 5 bits for red, setting the drawing colour to any value between 0 and 7 will result in the same, black, colour.

In addition to basic drawing commands, the GTT offers a number of more advanced features that enable complex graphical environments to be created with ease.

### Graphs

Graphs simplify the visual display of numerical information on the screen by taking care of the calculations and placement of graphics. Once a graph 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 graphs also allow for complex graphics to be used for more detailed rendering. Graph information is stored in a series of graph locations. A unique index is specified when a graph is created, and used to reference the graph in future commands.

## 1.3 Mass Storage

When powered with a jumper on the ”A” side of the mass storage jumper, the unit will appear to any PC as a mass storage device when connected via the Mass Storage USB header. The onboard SD memory card is used to store all user fonts, bitmaps, 9-slices, animations and start-up settings.

At run time, files can be loaded from the mass storage SD card into specific memory buffers for use as outlined below. Each font, bitmap, 9-slice, animation, label, and trace buffer has 256 unique index slots available, and up to 32MB of data can be stored across all buffers at any one time.

### Fonts

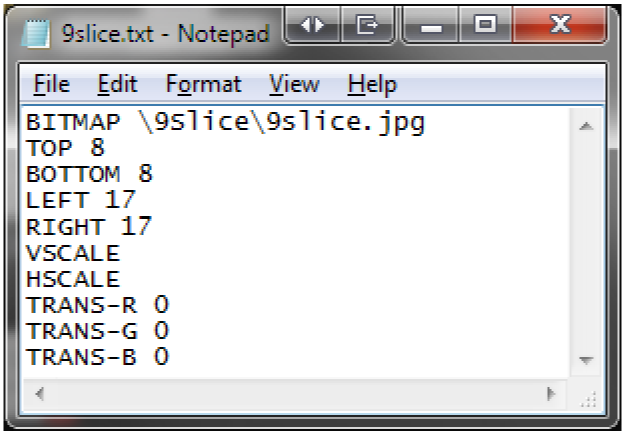
TrueType Font (.ttf) files can be uploaded to the SD memory card, loaded into a font buffer, and used to display customizable text. Commands are provided to modify the size, colour, and rendering of the loaded text. True Type Font files can be copied from your Windows Fonts folder or downloaded online.

### Bitmaps

Bitmaps typically display static graphic information, but can also be used to create dynamic touch regions or animations. Furthermore, a specific transparent colour can be selected via command. Please note that transparency in the images themselves is not supported; all bitmaps should have no more than 8 bits per channel and only 3 colour channels. Supported extensions are .bmp, .gif, .jpg, and .png.

### 9-slices

The 9-Slice file format is a simple text file that describes how to take a bitmap, and slice it into portions to allow simple and elegant scaling. A simple rounded button example would appear as follows:



```
BITMAP \9slice\9slice.jpg
TOP 8
BOTTOM 8
LEFT 17
RIGHT 17
VSCALE
HSCALE
TRANS-R 0
TRANS-G 0
TRANS-B 0
```

*Figure 2: 9-Slice File Example*

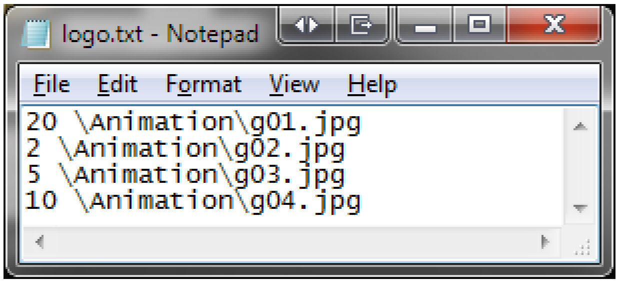
Each line must start with a keyword, followed by parameters. If a line contains an unrecognized keyword, the line is ignored. Keywords are defined in the Example Keypad Response table.

*Table 5: 9-slice Keywords*

| Keyword | Parameters | Description |
|---|---|---|
| BITMAP | 1 | Following the keyword, the bitmap that will be sliced is specified |
| TOP | 1 | Specifies how many pixels will be used from the top, for the top slice |
| BOTTOM | 1 | Specifies how many pixels will be used from the bottom, for the bottom slice |
| LEFT | 1 | Specifies how many pixels will be used from the left, for the left slice |
| RIGHT | 1 | Specifies how many pixels will be used from the right, for the right slice |
| VSCALE | 0 | If this keyword is present, when the 9-Slice is resized it will stretch the middle left and middle right slices to fill the space required. Without this keyword present, the tile will be repeated from the top down to fill the space. |
| HSCALE | 0 | If this keyword is present, when the 9-Slice is resized it will stretch the middle top and middle bottom slices to fill the space required. Without this keyword present, the tile will be repeated from the left to right to fill the space. |
| TRANS-R | 1 | The red component of the colour to make transparent in the 9-Slice |
| TRANS-G | 1 | The green component of the colour to make transparent in the 9-Slice |
| TRANS-B | 1 | The blue component of the colour to make transparent in the 9-Slice |

### Animations

While the image data for animations is stored in the Animation buffer, the actual state of animations is stored in a separate location. This allows animations to be removed from the screen, but remain in buffer memory. The animation descriptor file is a simple text file, with a series of lines of times to display a frame, and a path from the root directory to a bitmap file to use for that frame. For example:



```
20 \Animation\g01.jpg
2 \Animation\g02.jpg
5 \Animation\g03.jpg
10 \Animation\g04.jpg
```

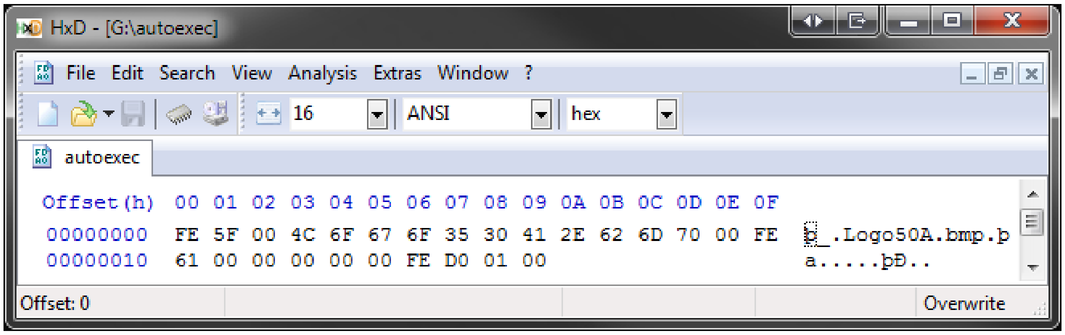
*Figure 3: Animation File Example*

The above example defines the first four frames of a sample logo animation stored in the Animation folder, with each frame displayed for 20ms, 2ms, 5ms, and 10ms respectively.

## 1.1 Scripting

### 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. The default autoexec file below, which ships from the factory, loads and displays a start screen before clearing the bitmap buffer.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  FE 5F 00 4C 6F 67 6F 35 30 41 2E 62 6D 70 00 FE  þ_.Logo50A.bmp.þ
00000010  61 00 00 00 00 00 FE D0 01 00                    a.....þÐ..
```

*Figure 4: Autoexec File Example*

Please note, if a command is started within the AUTOEXEC, however is not completed with the data in the AUTOEXEC, the command will wait for data from the serial port to complete the command. After which, the module will return to normal operations. The AUTOEXEC file is a special example of the script feature available on the GTT line.

## 1.2 Interactive Input

### Keypad

Unique values can be assigned for up to 25 keys. When a key event occurs it will be saved to a 20 key buffer. Key events will generate a return message that can be transmitted immediately or polled by toggling the auto transmit key press command. A sample return message is shown below.

*Table 6: Example Keypad Response*

| Prefix | Message ID | Length | | Event | Key ID |
|---|---|---|---|---|---|
| 252 | 165 | 0 | 2 | 0 | 65 |

Key presses will have a message ID of 165, and a data length equal to the number of bytes currently in the key buffer. Each key value will be preceded by an event byte as per the Keypad Event Types table.

*Table 7: Keypad Event Types*

| Value | 0 | 1 | 2 |
|---|---|---|---|
| Event | Press | Release | Repeat |

### Touch

Touch input***** allows the screen to return various types of up, down and move messages depending on the reporting style. Two distinct styles are available: region and coordinate. Both generate a return message with an identification number of 135, followed by event information.

*Table 8: Touch Event Types*

| Value | 0 | 1 | 2 |
|---|---|---|---|
| Event | Down | Up | Move |

In coordinate mode, the GTT will send an event type as listed above followed by signed short x and y coordinates of the touch location.

*Table 9: Example Co-ordinate Response*

| Prefix | Message ID | Length | | Event | X | | Y | |
|---|---|---|---|---|---|---|---|---|
| 252 | 135 | 0 | 5 | 0 | 0 | 50 | 0 | 10 |

In region mode, rectangular buttons are defined on the screen. When touch activities occur within regions, a visual update accompanies the event report listed in the Touch Event Types table. Events that occur outside defined regions may be reported as Region 255, when reporting is turned on.

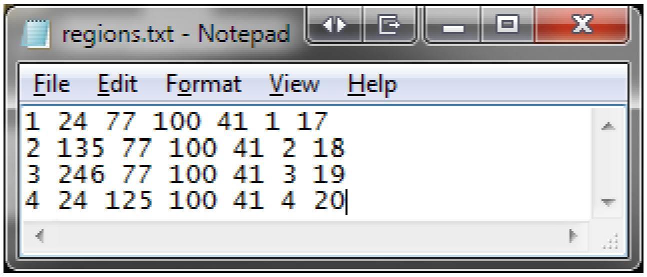
*Table 10: Example Region Response*

| Prefix | Message ID | Length | | Event | Region |
|---|---|---|---|---|---|
| 252 | 135 | 0 | 2 | 0 | 5 |

***Note:** Please ensure your touch screen is calibrated correctly by locating a touchcal.dat file within the SYSTEM folder of your SD Card. If necessary, calibration can be performed using the Calibrate Touch Screen command.

### Region File

Touch regions may be defined using a simple text file for speed and greater ease of use. Each line in a region file describes a single touch. There must be no leading blank spaces, only a single space between each field, and no trailing spaces. Bitmap buffers specified must be pre-loaded with desired images. An example of the first row of the calculator demo is shown below.



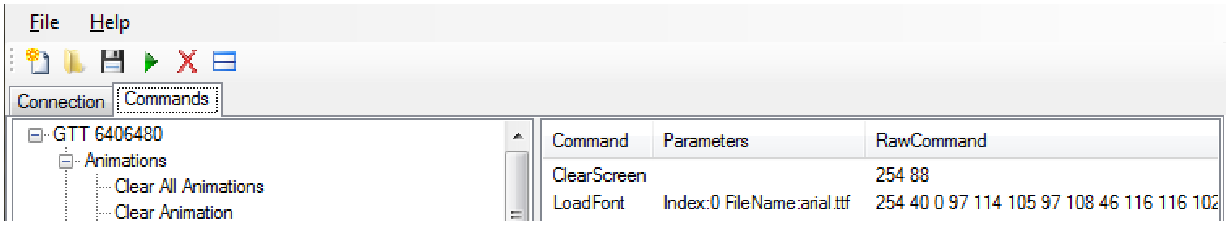
```
1 24 77 100 41 1 17
2 135 77 100 41 2 18
3 246 77 100 41 3 19
4 24 125 100 41 4 20
```

*Figure 5: Region File Example*

The file above would define four touch regions. The first has an index of 1 is positioned at coordinates (24, 77), a width of 100, and a height of 41. When it is pressed the bitmap in bitmap index 1 will be displayed, and when it is not pressed bitmap 17 will be displayed. Three similar regions follow this one.

## 1.3 Support Tools

Downloaded from http://www.matrixorbital.ca/software/, the GTT Project support tool provides a simple graphic interface with the full library of GTT series commands. This program allows users to drag and drop commands into a list that can be transmitted to the GTT, saved, and even loaded for later use.



| Command | Parameters | RawCommand |
|---|---|---|
| ClearScreen | | 254 88 |
| LoadFont | Index:0 FileName:arial.ttf | 254 40 0 97 114 105 97 108 46 116 116 102 |

*Figure 6: GTT Support Tools*

Command lists can also be converted into binary files using the save as feature. This will allow easy creation of an AUTOEXEC, or other script files. Finally, the support tool provides a debug window that will display the data flow to and from your GTT to ensure your command list executes correctly.

## 1.4 Firmware Upgrades

After release, Matrix Orbital may publish updates to the GTT code base or functionality that can be easily applied to the unit in the field. While in mass storage mode, place the latest upgrade.bin file from http://www.matrixorbital.ca/software/GTT2.0/ into the GTT upgrade folder. Then, cycle power to the unit, watch the screen for status updates, and wait for the screen to reboot. Finally, replace the GTT in your application and enjoy the new additions to the display you've come to know and love.

## 1.5 Application Notes

Full demonstration programs and code are available for many different Matrix Orbital displays in a number of different languages from the Application Note section at www.matrixorbital.ca/appnotes.

In addition, all files required to run the short examples described in the Advanced Features section are available for download from www.matrixorbital.ca/manuals/GTT_Series. Each example runs as an autoexec script and is described in the Instructions document.

Finally, a self-contained demo highlighting many of the features available in the GTT line is available at www.matrixorbital.ca/appnotes/GTT20FeatureDemo. No code is required as all functionality is provided through scripts. Simply copy the required files to your GTT to run the interactive demo.

For additional information regarding the features implemented, please see the Commands section below. If you have any questions please don't hesitate to contact a knowledgeable Matrix Orbital technical support representative.

# 2 Commands

## 2.1 Basic Commands

| 1.1 Set Customer Data | Dec | 254 52 | Length Data | 2.0 |
|---|---|---|---|---|
| | Hex | FE 34 | Length Data | |
| | ASCII | þ 4 | Length Data | |

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

| | | |
|---|---|---|
| Length | Byte | Length of the data to be transferred, in bytes. |
| Data | Byte(s) | Data to be written to the SD card. |

| 1.2 Get Customer Data | Dec | 254 53 | | 2.0 |
|---|---|---|---|---|
| | Hex | FE 35 | | |
| | ASCII | þ 5 | | |

Read data from the userdata.dat file in the \system\ folder of the SD card, previously written using the Write Customer Data command.

| | | |
|---|---|---|
| Return Message | 252 53 Length | Length Data |
| Length | Byte | Length of the data to be transferred, in bytes. |
| Data | Byte(s) | Data read from the SD Card. |

| 1.3 Set Backlight Brightness | Dec | 254 153 | Brightness | 2.0 |
|---|---|---|---|---|
| | Hex | FE 99 | Brightness | |
| | ASCII | þ ™ | Brightness | |

Set the brightness of the display backlight.

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

| 1.4 Get Backlight Brightness | Dec | 254 154 | | 2.0 |
|---|---|---|---|---|
| | Hex | FE 9A | | |
| | ASCII | þ š | | |

Read the current backlight brightness setting.

| | | |
|---|---|---|
| Return Message | 252 154 Length | Brightness |
| Brightness | Byte | The current backlight brightness. |

| 1.5 Write ScratchPad | Dec | 254 204 | Index Length Data | 2.0 |
|---|---|---|---|---|
| | Hex | FE CC | Index Length Data | |
| | ASCII | þ Ì | Index Length Data | |

Write data to volatile memory for temporary storage during operation.  Total memory size is 512 bytes.

| | | |
|---|---|---|
| Index | Short | Starting index of the data to be written. |
| Length | Short | Length of the data to be transferred, in bytes. |
| Data | Byte(s) | Data to temporarily save in volatile memory. |

| | | | | |
|---|---|---|---|---|
| **1.6 Read ScratchPad** | Dec | **254 205** | **Index Length** | **2.0** |
| | Hex | **FE CD** | **Index Length** | |
| | ASCII | **þ Í** | **Index Length** | |
| Read data previously stored in a specific section of volatile memory. | | | | |
| Index | **Short** | Starting index of the data to be read. | | |
| Length | **Short** | Length of the data requested. | | |
| **Return Message** | **252 205 Length** | Length Result | | |
| Length | **Short** | Length of the data to be transferred, in bytes. | | |
| Result | **Byte(s)** | Data read from specified location in volatile memory. | | |

## 2.2 Communication

| | | | |
|---|---|---|---|
| **2.1 Enter Mass Storage Mode** | Dec | **254 4** | **2.0** |
| | Hex | **FE 04** | |
| | ASCII | **þ [EOT]** | |
| Force the display to enter mass storage mode.  Useful for programmatically initiating a file transfer. | | | |

| | | | | |
|---|---|---|---|---|
| **2.2 Set Communication Channel** | Dec | **254 5** | **Channel** | **2.0** |
| | Hex | **FE 05** | **Channel** | |
| | ASCII | **þ [ENQ]** | **Channel** | |
| Set the default communication channel to be used for event driven data transmission; keypad or touch messages, for example.  The GTT will always answer user driven requests on the same channel they were transmitted. | | | | |
| Channel | **Byte** | Communication channel type, as per  eChannel values. | | |

*Table 11:  eChannel values*

| Value | Description |
|---|---|
| 0 | None |
| 1 | Serial |
| 2 | I2C |
| 3 | USB |
| 4 | CAN |
| 5 | SPI |
| 255 | Current |

| | | | | |
|---|---|---|---|---|
| **2.3 Set Baud Rate** | Dec | **254 57** | **BaudRate** | **2.0** |
| | Hex | **FE 39** | **BaudRate** | |
| | ASCII | **þ 9** | **BaudRate** | |
| Change the serial data rate used by the module. The change is implemented immediately after the last parameter byte has been received. Baud rate will reset to 115,200 on power up unless otherwise defined in the autoexec file. Serial command only. | | | | |
| BaudRate | **Integer** | The desired baud rate value. | | |

| 2.4 Set Flow Control Mode | Dec | 254 58 | FlowControl | 2.0 |
|---|---|---|---|---|
| | Hex | FE 3A | FlowControl | |
| | ASCII | þ : | FlowControl | |
| Toggle flow control between hardware and off settings.  Default is RTSCTS.  Serial command only. | | | | |
| FlowControl | Byte | Flow control setting, as per  eFlowControl values. | | |

*Table 12:  eFlowControl values*

| Value | Description |
|---|---|
| 0 | Off |
| 1 | RTSCTS |

| 2.5 Set I2C Address | Dec | 254 247 | I2Caddress | 2.0 |
|---|---|---|---|---|
| | Hex | FE F7 | I2Caddress | |
| | ASCII | þ ÷ | I2Caddress | |
| Immediately change the I2C write address.  Only even values are permitted as the next odd address will become the read address.  Default 8 bit address is 80 decimal (0x50 hex).  I2C command only. | | | | |
| I2Caddress | Byte | I2C write address, must be an even value. | | |

| 2.6 Echo | Dec | 254 255 | Message | 2.0 |
|---|---|---|---|---|
| | Hex | FE FF | Message | |
| | ASCII | þ ÿ | 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 | ASCII String | An arbitrary string that the module will return.  Limited to 4KB in length. | | |
| Return Message | 252 255 Length | ReturnMessage | | |
| ReturnMessage | ASCII String | The same arbitrary string originally sent. | | |

## 2.3 Module

| 3.1 Get Protocol Revision | Dec | 254 0 | 2.0 |
|---|---|---|---|
| | Hex | FE 00 | |
| | ASCII | þ [NUL] | |

Read the firmware version 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 on changes. For each command in this manual, the minimum firmware version required is listed at the top right.

| Return Message | 252 0 Length | Major Minor |
|---|---|---|
| Major | Byte | Major revision of the protocol used. |
| Minor | Byte | Minor revision of the protocol used. |

| 3.2 Reset Module | Dec | 254 1 | 2.0 |
|---|---|---|---|
| | Hex | FE 01 | |
| | ASCII | þ [SOH] | |

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

| 3.3 Delay | Dec | 254 2 | Time | 2.0 |
|---|---|---|---|---|
| | Hex | FE 02 | Time | |
| | ASCII | þ [STX] | Time | |

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

| | | |
|---|---|---|
| Time | Short | Length of delay in milliseconds. |

| 3.4 Get Display Metrics | Dec | 254 3 | 2.0 |
|---|---|---|---|
| | Hex | FE 03 | |
| | ASCII | þ [ETX] | |

Read the current width, height, and colour resolution of the display.

| Return Message | 252 3 Length | Width Height BitsRed BitsGreen BitsBlue |
|---|---|---|
| Width | Short | The width of the current display resolution in pixels. |
| Height | Short | The height of the current display resolution in pixels. |
| BitsRed | Byte | The number of bits used in the red channel. When less than 8 bits, byte length colour commands use the highest bits. |
| BitsGreen | Byte | The number of bits used in the green channel. When less than 8 bits, byte length colour commands use the highest bits. |
| BitsBlue | Byte | The number of bits used in the blue channel. When less than 8 bits, byte length colour commands use the highest bits. |

| 3.5 Get Module Version | Dec | 254 54 | 2.0 |
|---|---|---|---|
| | Hex | FE 36 | |
| | ASCII | þ 6 | |

Read the specific hardware revision of the module.

| Return Message | 252 54 Length | Major Minor |
|---|---|---|
| Major | Byte | Major revision of the hardware used. |
| Minor | Byte | Minor revision of the hardware used. |

| | | | |
|---|---|---|---|
| **3.6 Get Module Type** | Dec | **254 55** | **2.0** |
| | Hex | **FE 37** | |
| | ASCII | **þ 7** | |
| Read a two byte value that can be used to identify the display. | | | |
| **Return Message** | **252 55 Length** | Module | |
| Module | **Short** | The unique number of the module, as per  eModule values. | |

*Table 13:  eModule values*

| Value | Description |
|---|---|
| 37638 | GTT35A |
| 37648 | GTT38A |
| 37633 | GTT43A |
| 37634 | GTT50A |
| 37635 | GTT57A |
| 37636 | GTT70A |

| | | | |
|---|---|---|---|
| **3.7 Get Module String** | Dec | **254 56** | **2.0** |
| | Hex | **FE 38** | |
| | ASCII | **þ 8** | |
| Read the name of the module in string format. | | | |
| **Return Message** | **252 56 Length** | ModuleString | |
| ModuleString | **ASCII String** | The name of the module. | |

## 2.4 Drawing

| | | | | |
|---|---|---|---|---|
| **4.1 Set Background Drawing Colour** | Dec | **254 86** | **R G B** | **2.0** |
| | Hex | **FE 56** | **R G B** | |
| | ASCII | **þ V** | **R G B** | |
| Set the display colour when the Clear Screen command is sent. The default colour on startup is black. | | | | |
| R | **Byte** | Intensity of red, 0 to 255, limited to display metrics. | | |
| G | **Byte** | Intensity of green, 0 to 255, limited to display metrics. | | |
| B | **Byte** | Intensity of blue, 0 to 255, limited to display metrics. | | |

| | | | | |
|---|---|---|---|---|
| **4.2 Get Background Drawing Colour** | Dec | **254 87** | | **2.0** |
| | Hex | **FE 57** | | |
| | ASCII | **þ W** | | |
| Read the background colour that is currently being used by all drawing commands. | | | | |
| **Return Message** | **252 87 Length** | R G B | | |
| R | **Byte** | Intensity of red, 0 to 255, limited to display metrics. | | |
| G | **Byte** | Intensity of green, 0 to 255, limited to display metrics. | | |
| B | **Byte** | Intensity of blue, 0 to 255, limited to display metrics. | | |

| | | | | |
|---|---|---|---|---|
| **4.3 Clear Screen** | Dec | **254 88** | | **2.0** |
| | Hex | **FE 58** | | |
| | ASCII | **þ X** | | |
| Clear the screen, and reset the coordinates for both continued line and font insertion commands to zero. | | | | |

| | | | | |
|---|---|---|---|---|
| **4.4 Scroll Screen** | Dec | **254 89** | **X Y Width Height MoveX MoveY** | **2.0** |
| | Hex | **FE 59** | **X Y Width Height MoveX MoveY** | |
| | ASCII | **þ Y** | **X Y Width Height MoveX MoveY** | |
| Scroll the contents of a specified portion of the screen. | | | | |
| X | **Signed Short** | Leftmost coordinate of the scroll window. | | |
| Y | **Signed Short** | Topmost coordinate of the scroll window. | | |
| Width | **Signed Short** | Width of the scroll window. | | |
| Height | **Signed Short** | Height of the scroll window. | | |
| MoveX | **Signed Short** | Number of pixels to scroll horizontally. | | |
| MoveY | **Signed Short** | Number of pixels to scroll vertically. | | |

| | | | | |
|---|---|---|---|---|
| **4.5 Enable Manual Update** | Dec | **254 90** | **Enable** | **2.0** |
| | Hex | **FE 5A** | **Enable** | |
| | ASCII | **þ Z** | **Enable** | |

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

| | | |
|---|---|---|
| Enable | **Byte** | Desired manual update setting, as per  eEnable values. |

*Table 14:  eEnable values*

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

| | | | |
|---|---|---|---|
| **4.6 Manual Update** | Dec | **254 91** | **2.0** |
| | Hex | **FE 5B** | |
| | ASCII | **þ [** | |

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

| | | | | |
|---|---|---|---|---|
| **4.7 Flush Region** | Dec | **254 92** | **X Y Width Height** | **2.0** |
| | Hex | **FE 5C** | **X Y Width Height** | |
| | ASCII | **þ ** | **X Y Width Height** | |

Manually flush all graphic data in a specified region to the screen.

| | | |
|---|---|---|
| X | **Signed Short** | Leftmost coordinate of the flush window. |
| Y | **Signed Short** | Topmost coordinate of the flush window. |
| Width | **Signed Short** | Width of the flush window. |
| Height | **Signed Short** | Height of the flush window. |

| | | | | |
|---|---|---|---|---|
| **4.8 Set Drawing Colour** | Dec | **254 99** | **R G B** | **2.0** |
| | Hex | **FE 63** | **R G B** | |
| | ASCII | **þ c** | **R G B** | |

Set the colour to be used for all future drawing commands. The default colour is white.

| | | |
|---|---|---|
| R | **Byte** | Intensity of red, 0 to 255, limited to display metrics. |
| G | **Byte** | Intensity of green, 0 to 255, limited to display metrics. |
| B | **Byte** | Intensity of blue, 0 to 255, limited to display metrics. |

| 4.9 Get Drawing Colour | Dec | 254 100 | | 2.0 |
|---|---|---|---|---|
| | Hex | FE 64 | | |
| | ASCII | þ d | | |
| Read the colour that is currently being used by all drawing commands. | | | | |
| Return Message | 252 100 Length | R G B | | |
| R | Byte | Intensity of red, 0 to 255, limited to display metrics. | | |
| G | Byte | Intensity of green, 0 to 255, limited to display metrics. | | |
| B | Byte | Intensity of blue, 0 to 255, limited to display metrics. | | |

| 4.10 Continue Line | Dec | 254 101 | X Y | 2.0 |
|---|---|---|---|---|
| | Hex | FE 65 | X Y | |
| | ASCII | þ 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 Pixel, Draw Line, and Continue Line commands. | | | | |
| X | Signed Short | Horizontal coordinate of line terminus. | | |
| Y | Signed Short | Vertical coordinate of line terminus | | |

| 4.11 Draw Line | Dec | 254 108 | X1 Y1 X2 Y2 | 2.0 |
|---|---|---|---|---|
| | Hex | FE 6C | X1 Y1 X2 Y2 | |
| | ASCII | þ 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 Short | Horizontal coordinate of first line terminus. | | |
| Y1 | Signed Short | Vertical coordinate of first line terminus. | | |
| X2 | Signed Short | Horizontal coordinate of second line terminus. | | |
| Y2 | Signed Short | Vertical coordinate of second line terminus. | | |

| 4.12 Draw Pixel | Dec | 254 112 | X Y | 2.0 |
|---|---|---|---|---|
| | Hex | FE 70 | X Y | |
| | ASCII | þ p | X Y | |
| Draw a single pixel at the specified coordinate using the current drawing colour. | | | | |
| X | Signed Short | Horizontal position of pixel to be drawn. | | |
| Y | Signed Short | Vertical position of pixel to be drawn. | | |

| 4.13 Draw Rectangle | Dec | 254 114 | X Y Width Height | 2.0 |
|---|---|---|---|---|
| | Hex | FE 72 | X Y Width Height | |
| | ASCII | þ r | X Y Width Height | |
| Draw a rectangular frame one pixel wide using the current drawing colour. | | | | |
| X | Signed Short | Leftmost coordinate of the rectangle. | | |
| Y | Signed Short | Topmost coordinate of the rectangle. | | |
| Width | Short | Width of the rectangle. | | |
| Height | Short | Height of the rectangle. | | |

| 4.14 Draw Filled Rectangle | Dec | 254 120 | X Y Width Height | 2.0 |
|---|---|---|---|---|
| | Hex | FE 78 | X Y Width Height | |
| | ASCII | þ x | X Y Width Height | |

Draw a filled rectangle using the current drawing colour.

| | | |
|---|---|---|
| X | **Signed Short** | Leftmost coordinate of the rectangle. |
| Y | **Signed Short** | Topmost coordinate of the rectangle. |
| Width | **Short** | Width of the rectangle. |
| Height | **Short** | Height of the rectangle. |

| 4.15 Draw Circle | Dec | 254 123 | X Y Radius | 2.0 |
|---|---|---|---|---|
| | Hex | FE 7B | X Y Radius | |
| | ASCII | þ { | X Y Radius | |

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

| | | |
|---|---|---|
| X | **Signed Short** | Horizontal coordinate of circle centre. |
| Y | **Signed Short** | Vertical coordinate of circle centre. |
| Radius | **Short** | Radius of the circle. Diameter will be twice radius, plus one centre pixel. |

| 4.16 Draw Filled Circle | Dec | 254 124 | X Y Radius | 2.0 |
|---|---|---|---|---|
| | Hex | FE 7C | X Y Radius | |
| | ASCII | þ \| | X Y Radius | |

Draw a filled circle using the current drawing colour.

| | | |
|---|---|---|
| X | **Signed Short** | Horizontal coordinate of circle centre. |
| Y | **Signed Short** | Vertical coordinate of circle centre. |
| Radius | **Short** | Radius of the circle. Diameter will be twice radius, plus one centre pixel. |

| 4.17 Draw an Ellipse | Dec | 254 125 | X Y XRadius YRadius | 2.0 |
|---|---|---|---|---|
| | Hex | FE 7D | X Y XRadius YRadius | |
| | ASCII | þ } | X Y XRadius YRadius | |

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

| | | |
|---|---|---|
| X | **Signed Short** | Horizontal coordinate of ellipse centre. |
| Y | **Signed Short** | Vertical coordinate of ellipse centre. |
| XRadius | **Short** | Horizontal Radius of the ellipse. |
| YRadius | **Short** | Vertical Radius of the ellipse. |

| 4.18 Draw a Filled Ellipse | Dec | 254 126 | X Y XRadius YRadius | 2.0 |
|---|---|---|---|---|
| | Hex | FE 7E | X Y XRadius YRadius | |
| | ASCII | þ ~ | X Y XRadius YRadius | |

Draw a filled ellipse using the current drawing colour.

| | | |
|---|---|---|
| X | **Signed Short** | Horizontal coordinate of ellipse centre. |
| Y | **Signed Short** | Vertical coordinate of ellipse centre. |
| XRadius | **Short** | Horizontal Radius of the ellipse. |
| YRadius | **Short** | Vertical Radius of the ellipse. |

| | | | | |
|---|---|---|---|---|
| **4.19 Draw Rounded Rectangle** | Dec | **254 127** | **X Y Width Height Radius** | **2.0** |
| | Hex | **FE 7F** | **X Y Width Height Radius** | |
| | ASCII | **þ □** | **X Y Width Height Radius** | |

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 Short** | Leftmost coordinate of the rectangle |
| Y | **Signed Short** | Topmost coordinate of the rectangle |
| Width | **Signed Short** | Width of the rectangle. |
| Height | **Signed Short** | Height of the rectangle. |
| Radius | **Short** | Radius of the rounded corners. |

| | | | | |
|---|---|---|---|---|
| **4.20 Draw Filled Rounded Rectangle** | Dec | **254 128** | **X Y Width Height Radius** | **2.0** |
| | Hex | **FE 80** | **X Y Width Height Radius** | |
| | ASCII | **þ €** | **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 Short** | Leftmost coordinate of the rectangle |
| Y | **Signed Short** | Topmost coordinate of the rectangle |
| Width | **Signed Short** | Width of the rectangle. |
| Height | **Signed Short** | Height of the rectangle. |
| Radius | **Short** | Radius of the rounded corners. |

| | | | | |
|---|---|---|---|---|
| **4.21 Draw Triangle** | Dec | **254 129** | **X1 Y1 X2 Y2 X3 Y3** | **2.0** |
| | Hex | **FE 81** | **X1 Y1 X2 Y2 X3 Y3** | |
| | ASCII | **þ ⍰** | **X1 Y1 X2 Y2 X3 Y3** | |

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

| | | |
|---|---|---|
| X1 | **Signed Short** | Horizontal coordinate of the first point. |
| Y1 | **Signed Short** | Vertical coordinate of the first point. |
| X2 | **Signed Short** | Horizontal coordinate of the second point. |
| Y2 | **Signed Short** | Vertical coordinate of the second point. |
| X3 | **Signed Short** | Horizontal coordinate of the third point. |
| Y3 | **Signed Short** | Vertical coordinate of the third point. |

| | | | | |
|---|---|---|---|---|
| **4.22 Draw Filled Triangle** | Dec | **254 130** | **X1 Y1 X2 Y2 X3 Y3** | **2.0** |
| | Hex | **FE 82** | **X1 Y1 X2 Y2 X3 Y3** | |
| | ASCII | **þ ‚** | **X1 Y1 X2 Y2 X3 Y3** | |

Draw a filled triangle using the current drawing colour.

| | | |
|---|---|---|
| X1 | **Signed Short** | Horizontal coordinate of the first point. |
| Y1 | **Signed Short** | Vertical coordinate of the first point. |
| X2 | **Signed Short** | Horizontal coordinate of the second point. |
| Y2 | **Signed Short** | Vertical coordinate of the second point. |
| X3 | **Signed Short** | Horizontal coordinate of the third point. |
| Y3 | **Signed Short** | Vertical coordinate of the third point. |

## 2.5 Buffers

| | | | | |
|---|---|---|---|---|
| **5.1 Load Font** | Dec | **254 40** | **Index FileName** | **2.0** |
| | Hex | **FE 28** | **Index FileName** | |
| | ASCII | **þ (** | **Index FileName** | |
| Load a font file from the SD card into a font buffer for use. | | | | |
| Index | **Byte** | Index used to identify the font. Specific to fonts. | | |
| FileName | **ASCII String** | Filename, including path from the root folder, of the font file to load. | | |
| **Return Message** | **252 40 Length** | Result | | |
| Result | **Byte** | Outcome of Load Font command, as per  eErrorCode values. | | |

*Table 15:  eErrorCode values*

| Value | Description |
|---|---|
| 0 | File Not Found |
| 1 | Invalid Bitmap File Format |
| 2 | Invalid 9Slice Metrics |
| 3 | Invalid 9Slice 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 |
| 11 | Invalid Region File Format |
| 12 | Invalid Touch Calibration |
| 13 | Successful Touch Calibration |
| 14 | Invalid File Format |
| 15 | Invalid Trace Index |
| 16 | Invalid Touch Region |
| 254 | Success |
| 255 | Unknown Exception |

| | | | | |
|---|---|---|---|---|
| **5.2 Load Bitmap** | Dec | **254 95** | **Index FileName** | **2.0** |
| | Hex | **FE 5F** | **Index FileName** | |
| | ASCII | **þ _** | **Index FileName** | |
| Load a bitmap file from the SD card into a bitmap buffer for use.  Supported formats are BMP, GIF, JPG, and PNG (all in RGB format only, an alpha channel is not supported). | | | | |
| Index | **Byte** | Index used to identify the bitmap. Specific to bitmaps, and screen rectangles. | | |
| FileName | **ASCII String** | Filename, including path from the root folder, of the bitmap file to load. | | |
| **Return Message** | **252 95 Length** | Result | | |
| Result | **Byte** | Outcome of Load Bitmap command, as per  eErrorCode values. | | |

| | | | | |
|---|---|---|---|---|
| **5.3 Copy Screen Rectangle** | Dec | **254 96** | **Index X Y Width Height** | **2.0** |
| | Hex | **FE 60** | **Index X Y Width Height** | |
| | ASCII | **þ `** | **Index X Y Width Height** | |
| Save a rectangular region of the screen to a bitmap buffer, allowing it to be recalled at a later time. | | | | |
| Index | **Byte** | Index used to identify the screen section.  Specific to bitmaps and screen rectangles. | | |
| X | **Signed Short** | Leftmost coordinate. | | |
| Y | **Signed Short** | Topmost coordinate. | | |
| Width | **Short** | Width of the screen section. | | |
| Height | **Short** | Height of the screen section. | | |

| | | | | |
|---|---|---|---|---|
| **5.4 Load 9-Slice** | Dec | **254 144** | **Index Filename** | **2.0** |
| | Hex | **FE 90** | **Index Filename** | |
| | ASCII | **þ ⍰** | **Index Filename** | |
| Load a 9-slice file from the SD card into a 9-Slice buffer for use. | | | | |
| Index | **Byte** | Index used to identify the 9-slice.  Specific to 9-slices. | | |
| Filename | **ASCII String** | Filename, including path from the root folder, of the 9-Slice file to load. | | |
| **Return Message** | **252 144 Length** | Result | | |
| Result | **Byte** | Outcome of Load Nine Slice command, as per  eErrorCode values. | | |

| | | | | |
|---|---|---|---|---|
| **5.5 Load Animation** | Dec | **254 192** | **Index Filename** | **2.0** |
| | Hex | **FE C0** | **Index Filename** | |
| | ASCII | **þ À** | **Index Filename** | |
| Load an animation file from the SD card into an animation buffer for use.  Refer to the Animations entry in the Features section for more information. | | | | |
| Index | **Byte** | Index used to identify the animation.  Specific to animations. | | |
| Filename | **ASCII String** | Filename, including path from the root folder, of the animation file to load. | | |

| | | | | |
|---|---|---|---|---|
| **5.6 Clear  A Buffer** | Dec | **254 208** | **Type Index** | **2.0** |
| | Hex | **FE D0** | **Type Index** | |
| | ASCII | **þ Ð** | **Type Index** | |
| Clear all data from a specific index of the selected buffer type, freeing buffer memory.  Note, Labels and Traces save a background image to buffer memory upon initialization. | | | | |
| Type | **Byte** | Type of buffer to clear, as per  eBuffers values. | | |
| Index | **Byte** | Index of the file to be cleared from buffer memory. | | |

*Table 16:  eBuffers values*

| Value | Description |
|---|---|
| 0 | Animations |
| 1 | Bitmaps |
| 2 | NineSlices |
| 3 | Fonts |
| 4 | Labels |
| 5 | Traces |

|  |  |  |  |
|---|---|---|---|
| **5.7 Clear All Buffers** | Dec | **254 209** | **2.0** |
|  | Hex | **FE D1** |  |
|  | ASCII | **þ Ñ** |  |

Clear all data from all buffers, freeing significant buffer memory.

## 2.6 Text

| 6.1 Create a Label | Dec | 254 16 | Index X Y Width Height Rot VJst HJst Font R G B | 2.1 |
|---|---|---|---|---|
| | Hex | FE 10 | Index X Y Width Height Rot VJst HJst Font R G B | |
| | ASCII | þ [DLE] | Index X Y Width Height Rot VJst HJst Font R G B | |

Designate a portion of the screen that can easily be updated with one line of text; useful when displaying variables.

| | | |
|---|---|---|
| Index | Byte | Index used to identify the label. Specific to labels. |
| X | Signed Short | Leftmost coordinate of the label region. |
| Y | Signed Short | Topmost coordinate of the label region. |
| Width | Signed Short | Width of the label region in pixels. |
| Height | Signed Short | Height of the label region in pixels. |
| Rot | Signed Short | Rotation of the text within the label. |
| VJst | Byte | Vertical justification of text within the label, as per eFontAlignVertical values. |
| HJst | Byte | Horizontal justification of text within the label, as per eFontAlignHorizontal values. |
| Font | Byte | Font index of a previously loaded font to be used for the label. |
| R | Byte | Intensity of red, 0 to 255, used for label font colour. |
| G | Byte | Intensity of green, 0 to 255, used for label font colour. |
| B | Byte | Intensity of blue, 0 to 255, used for label font colour. |

*Table 17: eFontAlignVertical values*

| Value | Description |
|---|---|
| 0 | Top |
| 1 | Bottom |
| 2 | Center |

*Table 18: eFontAlignHorizontal values*

| Value | Description |
|---|---|
| 0 | Left |
| 1 | Right |
| 2 | Center |

| 6.2 Update a Label | Dec | 254 17 | Index Format Value | 2.1 |
|---|---|---|---|---|
| | Hex | FE 11 | Index Format Value | |
| | ASCII | þ | Index Format Value | |

Update a previously defined label with new text. Send a null character (empty string) to clear a label.

| | | |
|---|---|---|
| Index | Byte | Index previously assigned to a defined label. |
| Format | Byte | Format of the string, as per eLabelFormat values. |
| Value | String | New string to display within the label. The string must match the format specified and should be capitalized and one line in height. |

*Table 19: eLabelFormat values*

| Value | Description |
|---|---|
| 0 | ASCII |
| 1 | Unicode |
| 2 | UTF-8 |

| | | | | |
|---|---|---|---|---|
| **6.3 Print Unicode String** | Dec | **254 36** | Text | **2.0** |
| | Hex | **FE 24** | Text | |
| | ASCII | **þ $** | Text | |
| Print a Unicode formatted string on the screen, using the current font. | | | | |
| Length | **Short** | Length of the Unicode string to be transmitted, in characters. | | |
| Text | **Unicode String** | Unicode formatted string. | | |

| | | | | |
|---|---|---|---|---|
| **6.4 Print UTF-8 String** | Dec | **254 37** | Text | **2.0** |
| | Hex | **FE 25** | Text | |
| | ASCII | **þ %** | Text | |
| Print a UTF-8 formatted string on the screen, using the current font. | | | | |
| Text | **String** | UTF-8 formatted string. | | |

| | | | | |
|---|---|---|---|---|
| **6.5 Set Control Character Mode** | Dec | **254 38** | Mode | **2.0** |
| | Hex | **FE 26** | Mode | |
| | ASCII | **þ &** | Mode | |
| Set the behaviour of defined control characters. Default is Unix mode. | | | | |
| Mode | **Byte** | Desired control character mode, as per eControlCharacterMode values. | | |

*Table 20: eControlCharacterMode values*

| Value | Description |
|---|---|
| 0 | Unix |
| 1 | Windows |

| | | | |
|---|---|---|---|
| **6.6 Get Control Character Mode** | Dec | **254 39** | **2.0** |
| | Hex | **FE 27** | |
| | ASCII | **þ '** | |
| Read the current control character mode. | | | |
| **Return Message** | **252 39 Length** | Mode | |
| Mode | **Byte** | Current control character mode, as per eControlCharacterMode values. | |

| | | | | |
|---|---|---|---|---|
| **6.7 Get String Extents** | Dec | **254 42** | Text | **2.0** |
| | Hex | **FE 2A** | Text | |
| | ASCII | **þ *** | Text | |
| Read 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. Note, the baseline is not provided. | | | | |
| Text | **ASCII String** | String whose extents are desired. | | |
| **Return Message** | **252 42 Length** | Width Height | | |
| Width | **Short** | Width of the rendered string. | | |
| Height | **Short** | Height of the rendered string. Includes ascender and descender heights. | | |

| 6.8 Set Text Window | Dec | 254 43 | X Y Width Height | 2.0 |
|---|---|---|---|---|
| | Hex | FE 2B | X Y Width Height | |
| | ASCII | þ + | X Y Width Height | |
| Create an invisible rectangle to which future text and insertion commands will be confined. | | | | |
| X | Signed Short | Leftmost coordinate of the text window. | | |
| Y | Signed Short | Topmost coordinate of the text window. | | |
| Width | Short | Width of the text window. | | |
| Height | Short | Height of the text window. | | |

| 6.9 Get Text Window | Dec | 254 44 | 2.0 |
|---|---|---|---|
| | Hex | FE 2C | |
| | ASCII | þ , | |
| Read the extents of the current text window. | | | |
| Return Message | 252 44 Length | X Y Width Height | |
| X | Signed Short | Leftmost coordinate of the text window. | |
| Y | Signed Short | Topmost coordinate of the text window. | |
| Width | Short | Height of the text window. | |
| Height | Short | Height of the text window. | |

| 6.10 Reset Font | Dec | 254 45 | 2.0 |
|---|---|---|---|
| | Hex | FE 2D | |
| | ASCII | þ - | |
| Reset the font in slot 0 to the default GTT style with the last selected colour. | | | |

| 6.11 Set Text Colour | Dec | 254 46 | R G B | 2.0 |
|---|---|---|---|---|
| | Hex | FE 2E | R G B | |
| | ASCII | þ . | R G B | |
| Set the colour of all subsequent text rendered in the current font, current text and other fonts are not affected. | | | | |
| R | Byte | Intensity of red, 0 to 255, limited to display metrics. | | |
| G | Byte | Intensity of green, 0 to 255, limited to display metrics. | | |
| B | Byte | Intensity of blue, 0 to 255, limited to display metrics. | | |

| 6.12 Get Text Colour | Dec | 254 47 | 2.0 |
|---|---|---|---|
| | Hex | FE 2F | |
| | ASCII | þ / | |
| Read the current text colour of the font in use. | | | |
| Return Message | 252 47 Length | R G B | |
| R | Byte | Intensity of red, 0 to 255, limited to display metrics. | |
| G | Byte | Intensity of green, 0 to 255, limited to display metrics. | |
| B | Byte | Intensity of blue, 0 to 255, limited to display metrics. | |

| | | | | |
|---|---|---|---|---|
| 6.13 Get Font | Dec | 254 48 | | 2.0 |
| | Hex | FE 30 | | |
| | ASCII | þ 0 | | |
| Read the font index of the font currently in use. | | | | |
| Return Message | 252 48 Length | Index | | |
| Index | Byte | Font index used to identify the current font file in the font buffer. | | |

| | | | | |
|---|---|---|---|---|
| 6.14 Set Font | Dec | 254 49 | Index | 2.0 |
| | Hex | FE 31 | Index | |
| | ASCII | þ 1 | Index | |
| Set the current font to one that has been loaded into the font buffer. | | | | |
| Index | Byte | Font index used to identify the desired font file in the font buffer. | | |
| Return Message | 252 49 Length | Result | | |
| Result | Byte | Outcome of Set Current Font command, as per eErrorCode values. | | |

| | | | | |
|---|---|---|---|---|
| 6.15 Set Font Size | Dec | 254 51 | PtSize | 2.0 |
| | Hex | FE 33 | PtSize | |
| | ASCII | þ 3 | PtSize | |
| Set the point size of the current font. Default is 24 point. | | | | |
| PtSize | Byte | Desired point size for the current font. | | |

| | | | | |
|---|---|---|---|---|
| 6.16 Get Font Size | Dec | 254 61 | | 2.1 |
| | Hex | FE 3D | | |
| | ASCII | þ = | | |
| Get the point size of the current font. | | | | |
| Return Message | 252 61 Length | PtSize | | |
| PtSize | Byte | Implemented point size for the current font. | | |

| | | | | |
|---|---|---|---|---|
| 6.17 Go Home | Dec | 254 72 | | 2.0 |
| | Hex | FE 48 | | |
| | ASCII | þ H | | |
| Set the text insertion point to the upper leftmost corner of the current text window. | | | | |

| | | | | |
|---|---|---|---|---|
| 6.18 Set Text Insertion Point | Dec | 254 121 | X Y | 2.0 |
| | Hex | FE 79 | X Y | |
| | ASCII | þ y | X Y | |
| Define the upper left coordinate of the next text string to be displayed, clipped to the current text window. | | | | |
| X | Signed Short | Desired leftmost coordinate of the insertion point. | | |
| Y | Signed Short | Desired topmost coordinate of the insertion point. | | |

|  |  |  |  |
|---|---|---|---|
| **6.19 Get Text Insertion Point** | Dec | **254 122** | **2.0** |
|  | Hex | **FE 7A** |  |
|  | ASCII | **þ z** |  |

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

|  |  |  |
|---|---|---|
| **Return Message** | **252 122 Length** | X Y |
| X | **Signed Short** | Current leftmost coordinate of the insertion point. |
| Y | **Signed Short** | Current topmost coordinate of the insertion point. |

|  |  |  |  |  |
|---|---|---|---|---|
| **6.20 Set Font Rendering Style** | Dec | **254 211** | **RenderType** | **2.0** |
|  | Hex | **FE D3** | **RenderType** |  |
|  | ASCII | **þ Ó** | **RenderType** |  |

Set the font rendering style of the current font. Greyscale offers a more polished appearance at the cost of performance. Default is greyscale.

|  |  |  |
|---|---|---|
| RenderType | **Byte** | Rendertype, as per eFontRenderType values. |

*Table 21: eFontRenderType values*

| Value | Description |
|---|---|
| 0 | Grayscale |
| 1 | Monochrome |

|  |  |  |  |  |
|---|---|---|---|---|
| **6.21 Set Font Anchor Style** | Dec | **254 212** | **AnchorType** | **2.0** |
|  | Hex | **FE D4** | **AnchorType** |  |
|  | ASCII | **þ Ô** | **AnchorType** |  |

Set the anchoring style of the current font, used for positioning commands. Labels use BaseLine only. Default is UpperLeft.

|  |  |  |
|---|---|---|
| AnchorType | **Byte** | Type of anchor, as per eAnchorType values. |

*Table 22: eAnchorType values*

| Value | Description |
|---|---|
| 0 | UpperLeft |
| 1 | BaseLine |

## 2.7 Bitmaps

| | | | | |
|---|---|---|---|---|
| **7.1 Display Bitmap** | Dec | **254 97** | **Index X Y** | **2.0** |
| | Hex | **FE 61** | **Index X Y** | |
| | ASCII | **þ a** | **Index X Y** | |
| Display a bitmap previously loaded into a bitmap buffer. | | | | |
| Index | **Byte** | Index used to identify the desired file in the bitmap buffer. | | |
| X | **Signed Short** | Leftmost coordinate. | | |
| Y | **Signed Short** | Topmost coordinate. | | |
| **Return Message** | **252 97 Length** | Result | | |
| Result | **Byte** | Outcome of Display Bitmap command, as per eErrorCode values. | | |

| | | | | |
|---|---|---|---|---|
| **7.2 Set Bitmap Transparency** | Dec | **254 98** | **Index R G B** | **2.0** |
| | Hex | **FE 62** | **Index R G B** | |
| | ASCII | **þ b** | **Index R G B** | |
| Set the transparent colour for all future renderings of a specific bitmap index. Does not affect previously drawn versions of the specified bitmap. | | | | |
| Index | **Byte** | Index used to identify the desired file in the bitmap buffer. | | |
| R | **Byte** | Intensity of red, 0 to 255, limited to display metrics. | | |
| G | **Byte** | Intensity of green, 0 to 255, limited to display metrics. | | |
| B | **Byte** | Intensity of blue, 0 to 255, limited to display metrics. | | |
| **Return Message** | **252 98 Length** | Result | | |
| Result | **Byte** | Outcome of Set Bitmap Transparency command, as per eErrorCode values. | | |

## 2.8 NineSlices

| | | | | |
|---|---|---|---|---|
| **8.1 Display 9-Slice** | Dec | **254 145** | **Index X Y Width Height** | **2.0** |
| | Hex | **FE 91** | **Index X Y Width Height** | |
| | ASCII | **þ ‘** | **Index X Y Width Height** | |

Display a 9-slice previously loaded into the 9-slice buffer.

| | | |
|---|---|---|
| Index | **Byte** | Index used to identify the desired file in the 9-slice buffer. |
| X | **Signed Short** | Leftmost coordinate. |
| Y | **Signed Short** | Topmost coordinate. |
| Width | **Short** | Width of the 9-slice. |
| Height | **Short** | Height of the 9-slice. |

## 2.9 Animations

| | | | | |
|---|---|---|---|---|
| **9.1 Set Up Animation** | Dec | **254 193** | **Index Instance X Y** | **2.0** |
| | Hex | **FE C1** | **Index Instance X Y** | |
| | ASCII | **þ Á** | **Index Instance X Y** | |

Define a region of the screen to be used to display a loaded animation. If an animation is already in use at that index it will be overwritten. Use the start animation command to display and play the file.

| | | |
|---|---|---|
| Index | **Byte** | Index used to identify the desired file in the animation buffer. |
| Instance | **Byte** | Index used to identify the state of a specific animation instance. |
| X | **Signed Short** | Leftmost coordinate. |
| Y | **Signed Short** | Topmost coordinate. |

| | | | | |
|---|---|---|---|---|
| **9.2 Start/Stop Animation** | Dec | **254 194** | **Instance State** | **2.0** |
| | Hex | **FE C2** | **Instance State** | |
| | ASCII | **þ Â** | **Instance State** | |

Start or stop a specific animation. After it is started, an animation will loop until stopped.

| | | |
|---|---|---|
| Instance | **Byte** | Index of a previously defined Animation instance to be started/stopped. |
| State | **Byte** | Desired animation state, as per eAnimationState values. |

*Table 23: eAnimationState values*

| Value | Description |
|---|---|
| 0 | Paused |
| 1 | Playing |

| | | | | |
|---|---|---|---|---|
| **9.3 Set Animation Frame** | Dec | **254 195** | **Instance Frame** | **2.0** |
| | Hex | **FE C3** | **Instance Frame** | |
| | ASCII | **þ Ã** | **Instance 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.

| | | |
|---|---|---|
| Instance | **Byte** | Index of a previously defined Animation instance whose frame will be changed. |
| Frame | **Byte** | Number of the frame to be displayed. |

| | | | | |
|---|---|---|---|---|
| **9.4 Get Animation Frame** | Dec | **254 196** | **Instance** | **2.0** |
| | Hex | **FE C4** | **Instance** | |
| | ASCII | **þ Ä** | **Instance** | |

Read the current frame of the specified animation instance.

| | | |
|---|---|---|
| Instance | **Byte** | Animation instance upon which the current frame query will be carried out. |
| **Return Message** | **252 196 Length** | Frame |
| Frame | **Byte** | Current animation frame. |

| | | | | |
|---|---|---|---|---|
| 9.5 Stop All Animations | Dec | 254 198 | | 2.0 |
| | Hex | FE C6 | | |
| | ASCII | þ Æ | | |

Stop all currently running animations at their present frame.

| | | | | |
|---|---|---|---|---|
| 9.6 Clear Animation | Dec | 254 199 | Instance | 2.0 |
| | Hex | FE C7 | Instance | |
| | ASCII | þ Ç | Instance | |

Stop the specified animation instance at the current frame and remove it from the animation queue. The animation image data will remain loaded in the animation buffer and can be reused by issuing the setup command.

| | | |
|---|---|---|
| Instance | Byte | Index of specific animation instance to be deleted. |

| | | | |
|---|---|---|---|
| 9.7 Clear All Animations | Dec | 254 200 | 2.0 |
| | Hex | FE C8 | |
| | ASCII | þ È | |

Stop all animation instances at their current frames and remove them from the animation queue. Animation image data loaded into buffers will remain, and can be reused by issuing the setup command.

| | | | |
|---|---|---|---|
| 9.8 Resume All Animations | Dec | 254 201 | 2.0 |
| | Hex | FE C9 | |
| | ASCII | þ É | |

Resume all stopped animations from their present frame.

## 2.10 Graphs

| 10.1 List All Bargraphs | Dec | 254 102 | 2.0 |
|---|---|---|---|
| | Hex | FE 66 | |
| | ASCII | þ f | |
| Read a list of all bargraph entries. Three bytes per entry indicate current display use, type, and current value. | | | |
| Return Message | 252 102 Length | BarType BarValue | |
| BarType | Byte | Type of bargraph entry. | |
| BarValue | Signed Short | Current value of bargraph entry. | |

*Table 24: eBargraphType values*

| Value | Description |
|---|---|
| 0 | Unused |
| 1 | Plain |
| 2 | 9-Slice |

| 10.2 Define A Plain Bargraph | Dec | 254 103 | Index Min Max X Y Width Height FGR FGG FGB BGR BGG BGB D | 2.0 |
|---|---|---|---|---|
| | Hex | FE 67 | Index Min Max X Y Width Height FGR FGG FGB BGR BGG BGB D | |
| | ASCII | þ g | Index Min Max X Y Width Height FGR FGG FGB BGR BGG BGB D | |
| Define a region of the screen to be used to display a new plain bargraph identified by the specified index. New indices will overwrite old, invalid directions will default to 0, and inverted min and max values will be corrected. | | | | |
| Index | Byte | Index used to identify the desired bargraph. Specific to bargraphs. | | |
| Min | Signed Short | Minimum bargraph value. | | |
| Max | Signed Short | Maximum bargraph value. | | |
| X | Signed Short | Leftmost coordinate of the bargraph. | | |
| Y | Signed Short | Topmost coordinate of the bargraph. | | |
| Width | Signed Short | Width of the bargraph. | | |
| Height | Signed Short | Height of the bargraph. | | |
| FGR | Byte | Red component of the foreground colour. | | |
| FGG | Byte | Green component of the foreground colour. | | |
| FGB | Byte | Blue component of the foreground colour. | | |
| BGR | Byte | Red component of the background colour. | | |
| BGG | Byte | Green component of the background colour. | | |
| BGB | Byte | Blue component of the background colour. | | |
| D | Byte | Direction that the bargraph will take, as per eBargraphType values. | | |

*Table 25: eBargraphOrientation values*

| Value | Description |
|---|---|
| 0 | Bottom To Top |
| 1 | Left To Right |
| 2 | Right To Left |
| 3 | Top To Bottom |

| 10.3 Define A 9-Slice Bargraph | Dec | 254 104 | Index Min Max X Y Width Height BFG BBG D | 2.0 |
|---|---|---|---|---|
| | Hex | FE 68 | Index Min Max X Y Width Height BFG BBG D | |
| | ASCII | þ h | Index Min Max X Y Width Height BFG BBG D | |
| Define a region of the screen to be used to display a new 9-slice bargraph identified by the specified index. New indices will overwrite old, invalid directions will default to 0, and inverted min and max values will be corrected. | | | | |
| Index | Byte | Index used to identify the desired bargraph. Specific to bargraphs. | | |
| Min | Signed Short | Minimum bargraph value. | | |
| Max | Signed Short | Maximum bargraph value. | | |
| X | Signed Short | Leftmost coordinate of the bargraph. | | |
| Y | Signed Short | Topmost coordinate of the bargraph. | | |
| Width | Signed Short | Width of the bargraph. | | |
| Height | Signed Short | Height of the bargraph. | | |
| BFG | Byte | 9-Slice buffer index of the foreground image. | | |
| BBG | Byte | 9-Slice buffer index of the background image. | | |
| D | Byte | Direction that the bargraph will take, as per eBargraphOrientation values. | | |

| 10.4 Update A Bargraph Value | Dec | 254 105 | Index Value | 2.0 |
|---|---|---|---|---|
| | Hex | FE 69 | Index Value | |
| | ASCII | þ i | Index Value | |
| Update the value of the specified bargraph. Value will be bounded to the bargraph minimum and maximum. | | | | |
| Index | Byte | Index used to identify the desired bargraph. Specific to bargraphs. | | |
| Value | Signed Short | Current value of the bargraph. | | |
| Return Message | 252 105 Length | Result | | |
| Result | Byte | Outcome of Update A Bargraph Value command, as pereErrorCode values. | | |

| 10.5 Update Multiple Bargraph Values | Dec | 254 106 | Index Length Values | 2.0 |
|---|---|---|---|---|
| | Hex | FE 6A | Index Length Values | |
| | ASCII | þ j | Index Length Values | |
| Update the values of the specified bargraphs. Values will be bounded to the bargraph minimum and maximum. | | | | |
| Index | Byte | Index used to identify the first desired bargraph. Specific to bargraphs. | | |
| Length | Byte | Length of the data to be transferred, in bytes. | | |
| Values | Signed Short(s) | Current values, one for each bargraph index to be updated. | | |
| Return Message | 252 106 Length | Result | | |
| Result | Byte | Outcome of Set Multiple Bargraph Values command, as pereErrorCode values. | | |

| 10.6 Clear All Bargraphs | Dec | 254 107 | 2.0 |
|---|---|---|---|
| | Hex | FE 6B | |
| | ASCII | þ k | |
| Clear all bargraphs entries, erasing all attributes and setting all to an unused state. | | | |

|  |  |  |  |  |
|---|---|---|---|---|
| **10.7 Reset a Trace Value** | Dec | **254 109** | Index | **2.1** |
|  | Hex | **FE 6D** | Index |  |
|  | ASCII | **þ m** | Index |  |

Clear all data from a specified trace, and erase the trace area. As a result, the next Update Trace command behaves as though it is the very first update after initialization.

|  |  |  |
|---|---|---|
| Index | **Byte** | Index used to identify the desired trace. Specific to Traces. |

|  |  |  |  |  |
|---|---|---|---|---|
| **10.8 Reset Multiple Trace Values** | Dec | **254 110** | Index Number | **2.1** |
|  | Hex | **FE 6E** | Index Number |  |
|  | ASCII | **þ n** | Index Number |  |

Clear all data from multiple specified traces, and erase the trace areas. As a result, the next update trace commands behave as though they are the very first updates after initialization.

|  |  |  |
|---|---|---|
| Index | **Byte** | Index used to identify the first desired trace. Specific to Traces. |
| Number | **Byte** | Number of trace entries to be reset. |

|  |  |  |  |
|---|---|---|---|
| **10.9 List All Traces** | Dec | **254 115** | **2.1** |
|  | Hex | **FE 73** |  |
|  | ASCII | **þ s** |  |

Read a list of all traces. One entry is provided for each trace, an entry consists of the trace Index and a Value.

|  |  |  |
|---|---|---|
| **Return Message** | **252 115 Length** | ID Value |
| Index | **Byte** | Trace index number. One for each entry. 0 signifies an undefined entry. |
| Value | **Signed Short** | Current value of the trace. One for each entry. |

|  |  |  |  |  |
|---|---|---|---|---|
| **10.10 Initialize a Trace** | Dec | **254 116** | Index X Y Width Height Min Max Step Style Red Green Blue | **2.1** |
|  | Hex | **FE 74** | Index X Y Width Height Min Max Step Style Red Green Blue |  |
|  | ASCII | **þ t** | Index X Y Width Height Min Max Step Style Red Green Blue |  |

Initialize a graph trace for later implementation. Traces defined in the same area with the same step and style can be updated simultaneously, as a multi-variable graph, using the Update Multiple Traces command. Individual traces can be updated using the Update a Trace command.

|  |  |  |
|---|---|---|
| Index | **Byte** | Index used to identify the desired trace. Specific to Traces. |
| X | **Signed Short** | Leftmost coordinate of the trace region. |
| Y | **Signed Short** | Topmost coordinate of the trace region. |
| Width | **Signed Short** | Width of the trace region. |
| Height | **Signed Short** | Height of the trace region. |
| Min | **Signed Short** | Value displayed at the lowest point of the trace. |
| Max | **Signed Short** | Value displayed at the highest point of the trace. |
| Step | **Byte** | Number of pixels shifted when a trace is updated. |
| Style | **Byte** | Orientation and Direction of the trace, as per eTraceTypeandDirection values. A style is created by summing values of individual attributes. For example, a Line with a Bottom Left origin, Shifting right has a Style value of 129. |
| Red | **Byte** | Intensity of red for trace colour, 0 to 255, limited to display metrics. |
| Green | **Byte** | Intensity of green for trace colour, 0 to 255, limited to display metrics. |
| Blue | **Byte** | Intensity of blue for trace colour, 0 to 255, limited to display metrics. |

*Table 26: eTraceTypeandDirection values*

| Value | Description |
|---|---|
| 0 | Bar |
| 1 | Line |
| 2 | Step |
| 3 | Box |
| 0 | Bottom Left |
| 16 | Left Up |
| 32 | Top Right |
| 48 | Right Down |
| 64 | Bottom Right |
| 80 | Left Down |
| 96 | Top Left |
| 112 | Right Up |
| 0 | Shift Toward Origin |
| 128 | Shift Away From Origin |

| | | | | |
|---|---|---|---|---|
| **10.11 Update a Trace** | Dec | **254 117** | **Index Value** | **2.1** |
| | Hex | **FE 75** | **Index Value** | |
| | ASCII | **þ u** | **Index Value** | |
| Update the value of the trace at the specified index. Trace will be bounded to the minimum and maximum. | | | | |
| Index | **Byte** | Index used to identify the desired trace. Specific to Traces. | | |
| Value | **Signed Short** | Current value of the trace index to be updated. | | |

| | | | | |
|---|---|---|---|---|
| **10.12 Update Multiple Traces** | Dec | **254 118** | **Index Length Values** | **2.1** |
| | Hex | **FE 76** | **Index Length Values** | |
| | ASCII | **þ v** | **Index Length Values** | |
| Simultaneously update the values of numerous traces, useful for updating a multi-variable graph. | | | | |
| Index | **Byte** | Index used to identify the first desired trace. Specific to Traces. | | |
| Length | **Byte** | Number of trace values to be set. | | |
| Values | **Signed Short(s)** | Current values, one for each of the trace index to be updated. | | |
| **Return Message** | **252 118 Length** | Result | | |
| Result | **Byte** | Outcome of Update Multiple Traces command, as pereErrorCode values. | | |

| | | | |
|---|---|---|---|
| **10.13 Clear All Traces** | Dec | **254 119** | **2.1** |
| | Hex | **FE 77** | |
| | ASCII | **þ w** | |
| Clear all trace entries from buffer memory, erasing all attributes and images, then setting all to an unused state. | | | |

| | | | | |
|---|---|---|---|---|
| **10.14 Set Trace Min/Max** | Dec | **254 148** | **Index Min Max** | **2.2** |
| | Hex | **FE** | **Index Min Max** | |
| | ASCII | **þ** | **Index Min Max** | |
| Update the min and max values of the specified trace. Trace will update to new minimum and maximum values. | | | | |
| Index | **Byte** | Index used to identify the previously defined trace. Specific to Traces. | | |
| Min | **Signed Short** | Minimum value for the specified trace. | | |
| Max | **Signed Short** | Maximum value for the specified trace. | | |

| | | | | |
|---|---|---|---|---|
| **10.15 Get Trace Min/Max** | Dec | **254 149** | **Index** | **2.2** |
| | Hex | **FE** | **Index** | |
| | ASCII | **þ** | **Index** | |
| Get the current min and max values of the specified trace. | | | | |
| Index | **Byte** | Index used to identify the previously defined trace. Specific to Traces. | | |
| **Return Message** | **252 149 Length** | Min Max | | |
| Min | **Signed Short** | Current minimum value for the specified trace. | | |
| Max | **Signed Short** | Current maximum value for the specified trace. | | |

## 2.11 Keypad

| | | | |
|---|---|---|---|
| **11.1 Clear Key Buffer** | Dec | **254 69** | **2.0** |
| | Hex | **FE 45** | |
| | ASCII | **þ E** | |
| Clear all saved key presses from the key buffer. | | | |

| | | | | |
|---|---|---|---|---|
| **11.2 Set Keypad Transmit Mode** | Dec | **254 79** | **AutoTransmit** | **2.0** |
| | Hex | **FE 4F** | **AutoTransmit** | |
| | ASCII | **þ O** | **AutoTransmit** | |
| Toggle auto transmission of key values. Can be used to poll the key buffer. | | | | |
| AutoTransmit | **Byte** | Auto transmit mode, as per eOnOff values. | | |

*Table 27: eOnOff values*

| Value | Description |
|---|---|
| 0 | Off |
| 1 | On |

| | | | | |
|---|---|---|---|---|
| **11.3 Set Debounce Time** | Dec | **254 85** | **Mode** | **2.0** |
| | Hex | **FE 55** | **Mode** | |
| | ASCII | **þ U** | **Mode** | |
| Set the time, in milliseconds, between a key press and a key read by the display. Most switches will bounce when pressed; the debounce time allows the switch to settle for an accurate read. Default is 64ms. | | | | |
| Mode | **Byte** | Debounce time in milliseconds. | | |

| | | | | |
|---|---|---|---|---|
| **11.4 Set Typematic Interval** | Dec | **254 158** | **Interval** | **2.0** |
| | Hex | **FE 9E** | **Interval** | |
| | ASCII | **þ ž** | **Interval** | |
| Set the interval between reported key presses when a key is held. Typematic mode only. | | | | |
| Interval | **Short** | Time between key reports, in milliseconds, default is 200ms. | | |

| | | | | |
|---|---|---|---|---|
| **11.5 Set Typematic Delay** | Dec | **254 159** | **Delay** | **2.0** |
| | Hex | **FE 9F** | **Delay** | |
| | ASCII | **þ Ÿ** | **Delay** | |
| Set the delay between the first key press and first typematic report when a key is held. Typematic mode only. | | | | |
| Delay | **Short** | Time key must be held to trigger a report, in milliseconds, default is 1000ms. | | |

### 11.6 Set Auto Repeat Mode

| | | | |
|---|---|---|---|
| Dec | 254 165 | Mode | 2.0 |
| Hex | FE A5 | Mode | |
| ASCII | þ ¥ | Mode | |

Set key press repeat mode to typematic or hold.  In typematic mode if a key press is held, by default the key value is transmitted immediately, then 5 times a second after a 1 second delay.  In hold mode, the key down value is transmitted once when pressed, and then the key up value is sent when the key is released.  Default is Off.

| | | |
|---|---|---|
| Mode | Byte | Desired keypad auto repeat mode, as per eKeypadRepeatMode values. |

*Table 28:  eKeypadRepeatMode values*

| Value | Description |
|---|---|
| 0 | Off |
| 1 | Hold |
| 2 | Typematic |

### 11.7 Assign Keypad Codes

| | | |
|---|---|---|
| Dec | 254 213 | Length KeyCodes |
| Hex | FE D5 | Length KeyCodes |
| ASCII | þ Õ | Length KeyCodes |

Assign values reported to the host when a key press is detected.  Up to 25 keys may be defined.

| | | |
|---|---|---|
| Length | Byte | Length of the data to be transferred, in bytes. |
| KeyCodes | Byte(s) | A list of byte values for each key to be defined.  Default values are 65 through 90. |

## 2.12 Touch

| 12.1 Create A Touch Region | Dec | 254 132 | Index X Y Width Height Up Down | 2.0 |
|---|---|---|---|---|
| | Hex | FE 84 | Index X Y Width Height Up Down | |
| | ASCII | þ „ | Index X Y Width Height Up Down | |
| Create a region of the screen that responds to touch events with a defined single byte and visual update. | | | | |
| Index | Byte | Index used to identify the desired touch region. | | |
| X | Signed Short | Leftmost coordinate of the touch region. | | |
| Y | Signed Short | Topmost coordinate of the touch region. | | |
| Width | Short | Width of the touch region. | | |
| Height | Short | Height of the touch region. | | |
| Up | Byte | Index of a loaded bitmap that is displayed when the region is released. | | |
| Down | Byte | Index of a loaded bitmap that is displayed when the region is touched. | | |

| 12.2 Clear A Touch Region | Dec | 254 133 | Index | 2.0 |
|---|---|---|---|---|
| | Hex | FE 85 | Index | |
| | ASCII | þ … | Index | |
| Clear the specified touch region entry from memory and screen, ensuring its events will no longer be reported. | | | | |
| Index | Byte | Index used to identify the desired touch region. | | |

| 12.3 Clear All Touch Regions | Dec | 254 134 | | 2.0 |
|---|---|---|---|---|
| | Hex | FE 86 | | |
| | ASCII | þ † | | |
| Clear all touch regions, ensuring their touch events will no longer be reported. | | | | |

| 12.4 Change Touch Reporting Style | Dec | 254 135 | ReportingType | 2.0 |
|---|---|---|---|---|
| | Hex | FE 87 | ReportingType | |
| | ASCII | þ ‡ | ReportingType | |
| Customize the way in which touch events are reported. Default is Region Down. | | | | |
| ReportingType | Byte | Desired touch reporting style, as per eTouchReportingType values. | | |

*Table 29: eTouchReportingType values*

| Value | Description | Value | Description |
|---|---|---|---|
| 0 | Region None | 8 | Coord None |
| 1 | Region Down | 9 | Coord Down |
| 2 | Region Up | 10 | Coord Up |
| 3 | Region Up/Down | 11 | Coord Up/Down |
| 4 | Region Move | 12 | Coord Move |
| 5 | Region Move/Down | 13 | Coord Move/Down |
| 6 | Region Move/Up | 14 | Coord Move/Up |
| 7 | Region Move/Up/Down | 15 | Coord Move/Up/Down |

|  |  |  |  |  |
|---|---|---|---|---|
| **12.5 Set Dragging Threshold** | Dec | **254 137** | **Threshold** | **2.0** |
|  | Hex | **FE 89** | **Threshold** |  |
|  | ASCII | **þ ‰** | **Threshold** |  |
| Set the distance a press is required to travel before a move event is reported. Precision will vary inversely to data transmitted; care should be taken to find a suitable balance. Distance is calculated as [Δx]^2 + [Δy]^2 = d^2. |  |  |  |  |
| Threshold | **Short** | Dragging threshold value. Default is 3 pixels. |  |  |

|  |  |  |  |  |
|---|---|---|---|---|
| **12.6 Calibrate Touch Screen** | Dec | **254 139** |  | **2.0** |
|  | Hex | **FE 8B** |  |  |
|  | ASCII | **þ ‹** |  |  |
| Initiate the touch screen calibration sequence. After user input is complete a confirmation byte will be returned, new calibration settings will be loaded, and the calibration will be saved as \SYSTEM\touchcal.dat. Calibration can be restored from the file listed above at any time. |  |  |  |  |
| **Return Message** | **252 139 Length** | Result |  |  |
| Result | **Byte** | Outcome of Calibrate Touch Screen command, as per eCalibrationErrorCode values. |  |  |

*Table 30: eCalibrationErrorCode values*

| Value | Description |
|---|---|
| 1 | Calibration Successful |
| 12 | Calibration Invalid |

|  |  |  |  |  |
|---|---|---|---|---|
| **12.7 Load Region File** | Dec | **254 140** | **FileName** | **2.0** |
|  | Hex | **FE 8C** | **FileName** |  |
|  | ASCII | **þ Œ** | **FileName** |  |
| Load a group of touch region definitions from a file, old regions are cleared. See the Region File example. |  |  |  |  |
| FileName | **ASCII String** | Filename, and path from the root folder, of the region file to load. |  |  |
| **Return Message** | **252 140 Length** | Result |  |  |
| Result | **Byte** | Outcome of Load Region File command, as per eErrorCode values. |  |  |

|  |  |  |  |  |
|---|---|---|---|---|
| **12.8 Restore Touch Calibration** | Dec | **254 141** |  | **2.0** |
|  | Hex | **FE 8D** |  |  |
|  | ASCII | **þ ⍰** |  |  |
| Restore touch calibration using the data from \SYSTEM\touchcal.dat, if this file is present. |  |  |  |  |
| **Return Message** | **252 141 Length** | Result |  |  |
| Result | **Byte** | Outcome of Restore Touch Calibration command, as per eRestoreCalibrationErrorCode values. |  |  |

*Table 31: eRestoreCalibrationErrorCode values*

| Value | Description |
|---|---|
| 0 | Restore Calibration Invalid |
| 1 | Restore Calibration Successful |

| | | | | |
|---|---|---|---|---|
| **12.9 Set Out of Region Setting** | Dec | **254 142** | **Setting** | **2.0** |
| | Hex | **FE 8E** | **Setting** | |
| | ASCII | **þ Ž** | **Setting** | |
| Set reporting status for out of region touch responses. | | | | |
| Setting | **Byte** | Desired out of region setting, as per eOnOff values. Default is Off. | | |

| | | | | |
|---|---|---|---|---|
| **12.10 Get Out of Region Setting** | Dec | **254 143** | | **2.0** |
| | Hex | **FE 8F** | | |
| | ASCII | **þ ⍰** | | |
| Read the current out of region setting. | | | | |
| **Return Message** | **252 143 Length** | Report | | |
| Report | **Byte** | Current out of region setting, as per eOnOff values. | | |

| | | | | |
|---|---|---|---|---|
| **12.11 Set Region Activation State** | Dec | **254 146** | **Index State** | **2.2** |
| | Hex | **FE 92** | **Index State** | |
| | ASCII | **þ Æ** | **Index State** | |
| Set the activation state for a specific region. When a region is created, the default is activated. | | | | |
| Index | **Byte** | Region index used to identify an existing touch region. | | |
| State | **Byte** | Activation state, as per eEnable values. | | |

| | | | | |
|---|---|---|---|---|
| **12.12 Get Region Activation State** | Dec | **254 147** | **Index** | **2.2** |
| | Hex | **FE 93** | **Index** | |
| | ASCII | **þ ô** | **Index** | |
| Get the current activation state of a specific region. An invalid touch region error will be returned if the specified index does not contain a previously defined region. | | | | |
| Index | **Byte** | Region index used to identify an existing touch region. | | |
| **Return Message** | **252 147 Length** | State | | |
| State | **Byte** | Current activation state, as per eEnable values. | | |

## 2.13 Output

| | | | | |
|---|---|---|---|---|
| **13.1 Set GPO State** | Dec | **254 73** | **Number Setting** | **2.0** |
| | Hex | **FE 49** | **Number Setting** | |
| | ASCII | **þ I** | **Number Setting** | |

| | | |
|---|---|---|
| Toggle the specified GPO on or off, sourcing up to 15mA current at 5V per GPO or sinking to ground. | | |
| Number | **Byte** | GPO to be controlled. |
| Setting | **Byte** | GPO state, as per eGPOSetting values. |

*Table 32: eGPOSetting values*

| Value | Description |
|---|---|
| 1 | On |
| 0 | Off |

| | | | | |
|---|---|---|---|---|
| **13.2 Activate Motor** | Dec | **254 160** | **Frequency Duration** | **2.0** |
| | Hex | **FE A0** | **Frequency Duration** | |
| | ASCII | **þ** | **Frequency Duration** | |

| | | |
|---|---|---|
| Generate a vibration from the motor at the specified frequency for the defined duration. | | |
| Frequency | **Short** | Frequency of the vibration in Hertz. |
| Duration | **Short** | Duration of the vibration in milliseconds. |

| | | | | |
|---|---|---|---|---|
| **13.3 Set Input Feedback** | Dec | **254 182** | **InputOutputType DownFrequency UpFrequency** | **2.0** |
| | Hex | **FE B6** | **InputOutputType DownFrequency UpFrequency** | |
| | ASCII | **þ ¶** | **InputOutputType DownFrequency UpFrequency** | |

| | | |
|---|---|---|
| Initiate autonomous feedback by specifying a 50 millisecond output event for specific input events. | | |
| InputOutputType | **Byte** | Desired input event and output response types, as per eKeypadInputOutputType values. Multiple events and/or responses can be selected by summing values. |
| DownFrequency | **Short** | Frequency of the down event in Hertz. |
| UpFrequency | **Short** | Frequency of the up event in Hertz. |

*Table 33: eKeypadInputOutputType values*

| Value | Description |
|---|---|
| 0 | None |
| 1 | Output Beep |
| 2 | Output Motor |
| 4 | Input Keypad |
| 8 | Input Touch |

| | | | | |
|---|---|---|---|---|
| **13.4 Activate Buzzer and Motor** | Dec | **254 183** | **Frequency Duration** | **2.1** |
| | Hex | **FE B7** | **Frequency Duration** | |
| | ASCII | **þ ·** | **Frequency Duration** | |

| | | |
|---|---|---|
| Generate a vibration from the motor and a tone from the piezo buzzer simultaneously, at the specified frequency for the defined interval. | | |
| Frequency | **Short** | Frequency of the beep and vibration in Hertz. |
| Duration | **Short** | Duration of the beep in milliseconds. |

|  |  |  |  |  |
|---|---|---|---|---|
| **13.5 Activate Buzzer** | Dec | **254 187** | **Frequency Duration** | **2.0** |
|  | Hex | **FE BB** | **Frequency Duration** |  |
|  | ASCII | **þ »** | **Frequency Duration** |  |
| Generate a tone from the piezo buzzer at the specified frequency for the defined duration. |  |  |  |  |
| Frequency | **Short** | Frequency of the beep in Hertz. |  |  |
| Duration | **Short** | Duration of the beep in milliseconds. |  |  |

|  |  |  |  |  |
|---|---|---|---|---|
| **13.6 Set Default Buzzer Beep** | Dec | **254 188** | **Frequency Duration** | **2.0** |
|  | Hex | **FE BC** | **Frequency Duration** |  |
|  | ASCII | **þ ¼** | **Frequency Duration** |  |
| Set the frequency and duration of the default beep transmitted when the bell character is transmitted. |  |  |  |  |
| Frequency | **Short** | Frequency of the beep in Hertz. |  |  |
| Duration | **Short** | Duration of the beep in milliseconds. |  |  |

## 2.14 Scripts

| 14.1 Run Script File | Dec | 254 93 | FileName | 2.0 |
|---|---|---|---|---|
| | Hex | FE 5D | FileName | |
| | ASCII | þ ] | FileName | |

Send an array of bytes saved in the specified script file as if it was received from the serial port. Sending data to the serial port is still possible, but it will queue up in the receive buffer and will only be parsed after the execution of the script file. Scripts may be stacked up to 10 deep.

| | | |
|---|---|---|
| FileName | ASCII String | Filename, and path from the root folder, of the script file to run. |

| 14.2 Create a Scripted Region | Dec | 254 131 | ID X Y W H UpBitmap DownBitmap UpScript DownScript | 2.1 |
|---|---|---|---|---|
| | Hex | FE 83 | ID X Y W H UpBitmap DownBitmap UpScript DownScript | |
| | ASCII | þ ƒ | ID X Y W H UpBitmap DownBitmap UpScript DownScript | |

Create a region that responds to a touch event by executing scripts, in addition to the normal visual and event updates. Scripts always execute, regardless of the current touch reporting style. If only one script is desired, use an empty string for the other script filename.

| | | |
|---|---|---|
| ID | Byte | Index used to identify the desired touch region. |
| X | Signed Short | Leftmost coordinate of the scripted touch region. |
| Y | Signed Short | Topmost coordinate of the scripted touch region. |
| W | Short | Width of the scripted touch region. |
| H | Short | Height of the scripted touch region. |
| UpBitmap | Byte | Index of the loaded bitmap displayed when the region is released. |
| DownBitmap | Byte | Index of the loaded bitmap displayed when the region is pressed. |
| UpScript | ASCII String | Filename of the script to be executed when the region is released. |
| DownScript | ASCII String | Filename of the script to be executed when the region is pressed. |

| 14.3 Create a Scripted Key | Dec | 254 138 | ID Row Col UpScript DownScript | 2.1 |
|---|---|---|---|---|
| | Hex | FE 8A | ID Row Col UpScript DownScript | |
| | ASCII | þ Š | ID Row Col UpScript DownScript | |

Designate a specific key that responds to a press event by executing scripts. Scripts always execute, regardless of the current key reporting style. If only one script is desired, use an empty string for the other script filename.

| | | |
|---|---|---|
| ID | Byte | Index used to identify the desired key value. |
| Row | Byte | Row index of the scripted key. |
| Col | Byte | Column index of the scripted key. |
| UpScript | ASCII String | Filename of the script to be executed when the key is released. |
| DownScript | ASCII String | Filename of the script to be executed when the key is pressed. |

# 3 Appendix

## 3.1 Command Summary

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

*Table 34: Basic Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Set Customer Data | 52 | 34 | 4 | Length, Data | None |
| Get Customer Data | 53 | 35 | 5 | None | Length, Data |
| Set Backlight Brightness | 153 | 99 | ™ | Brightness | None |
| Get Backlight Brightness | 154 | 9A | š | None | Brightness |
| Write ScratchPad | 204 | CC | Ì | Index, Length, Data | None |
| Read ScratchPad | 205 | CD | Í | Index, Length | Length, Result |

*Table 35: Communication Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Enter Mass Storage Mode | 4 | 04 | [EOT] | None | None |
| Set Communication Channel | 5 | 05 | [ENQ] | Channel | None |
| Set Baud Rate | 57 | 39 | 9 | BaudRate | None |
| Set Flow Control Mode | 58 | 3A | : | FlowControl | None |
| Set I2C Address | 247 | F7 | ÷ | I2Caddress | None |
| Echo | 255 | FF | ÿ | Message | ReturnMessage |

*Table 36: Module Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Get Protocol Revision | 0 | 00 | [NUL] | None | Major, Minor |
| Reset Module | 1 | 01 | [SOH] | None | None |
| Delay | 2 | 02 | [STX] | Time | None |
| Get Display Metrics | 3 | 03 | [ETX] | None | Width, Height, BitsRed, BitsGreen, BitsBlue |
| Get Module Version | 54 | 36 | 6 | None | Major, Minor |
| Get Module Type | 55 | 37 | 7 | None | Module |
| Get Module String | 56 | 38 | 8 | None | ModuleString |

*Table 37: Drawing Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Set Background Drawing Colour | 86 | 56 | V | R, G, B | None |
| Get Background Drawing Colour | 87 | 57 | W | None | R, G, B |
| Clear Screen | 88 | 58 | X | None | None |
| Scroll Screen | 89 | 59 | Y | X, Y, Width, Height, MoveX, MoveY | None |
| Enable Manual Update | 90 | 5A | Z | Enable | None |
| Manual Update | 91 | 5B | [ | None | None |
| Flush Region | 92 | 5C | \ | X, Y, Width, Height | None |
| Set Drawing Colour | 99 | 63 | c | R, G, B | None |
| Get Drawing Colour | 100 | 64 | d | None | R, G, B |
| Continue Line | 101 | 65 | e | X, Y | None |
| Draw Line | 108 | 6C | l | X1, Y1, X2, Y2 | None |
| Draw Pixel | 112 | 70 | p | X, Y | None |
| Draw Rectangle | 114 | 72 | r | X, Y, Width, Height | None |
| Draw Filled Rectangle | 120 | 78 | x | X, Y, Width, Height | None |
| Draw Circle | 123 | 7B | { | X, Y, Radius | None |
| Draw Filled Circle | 124 | 7C | \| | X, Y, Radius | None |
| Draw an Ellipse | 125 | 7D | } | X, Y, XRadius, YRadius | None |
| Draw a Filled Ellipse | 126 | 7E | ~ | X, Y, XRadius, YRadius | None |
| Draw Rounded Rectangle | 127 | 7F | □ | X, Y, Width, Height, Radius | None |
| Draw Filled Rounded Rectangle | 128 | 80 | € | X, Y, Width, Height, Radius | None |
| Draw Triangle | 129 | 81 | ⍰ | X1, Y1, X2, Y2, X3, Y3 | None |
| Draw Filled Triangle | 130 | 82 | ‚ | X1, Y1, X2, Y2, X3, Y3 | None |

*Table 38: Buffers Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Load Font | 40 | 28 | ( | Index, FileName | Result |
| Load Bitmap | 95 | 5F | _ | Index, FileName | Result |
| Copy Screen Rectangle | 96 | 60 | ` | Index, X, Y, Width, Height | None |
| Load 9-Slice | 144 | 90 | ⍰ | Index, Filename | Result |
| Load Animation | 192 | C0 | À | Index, Filename | None |
| Clear A Buffer | 208 | D0 | Ð | Type, Index | None |
| Clear All Buffers | 209 | D1 | Ñ | None | None |

*Table 39: Text Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Create a Label | 16 | 10 | [DLE] | Index, X, Y, Width, Height, Rot, VJst, HJst, Font, R, G, B | None |
| Update a Label | 17 | 11 | | Index, Format, Value | None |
| Print Unicode String | 36 | 24 | $ | Length, Text | None |
| Print UTF-8 String | 37 | 25 | % | Text | None |
| Set Control Character Mode | 38 | 26 | & | Mode | None |
| Get Control Character Mode | 39 | 27 | ' | None | Mode |
| Get String Extents | 42 | 2A | * | Text | Width, Height |
| Set Text Window | 43 | 2B | + | X, Y, Width, Height | None |
| Get Text Window | 44 | 2C | , | None | X, Y, Width, Height |
| Reset Font | 45 | 2D | - | None | None |
| Set Text Colour | 46 | 2E | . | R, G, B | None |
| Get Text Colour | 47 | 2F | / | None | R, G, B |
| Get Font | 48 | 30 | 0 | None | Index |
| Set Font | 49 | 31 | 1 | Index | Result |
| Set Font Size | 51 | 33 | 3 | PtSize | None |
| Get Font Size | 61 | 3D | = | None | PtSize |
| Go Home | 72 | 48 | H | None | None |
| Set Text Insertion Point | 121 | 79 | y | X, Y | None |
| Get Text Insertion Point | 122 | 7A | z | None | X, Y |
| Set Font Rendering Style | 211 | D3 | Ó | RenderType | None |
| Set Font Anchor Style | 212 | D4 | Ô | AnchorType | None |

*Table 40: Bitmaps Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Display Bitmap | 97 | 61 | a | Index, X, Y | Result |
| Set Bitmap Transparency | 98 | 62 | b | Index, R, G, B | Result |

*Table 41: NineSlices Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Display 9-Slice | 145 | 91 | ‘ | Index, X, Y, Width, Height | None |

*Table 42: Animations Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Set Up Animation | 193 | C1 | Á | Index, AnimationNum, X, Y | None |
| Start/Stop Animation | 194 | C2 | Â | AnimationNum, State | None |
| Set Animation Frame | 195 | C3 | Ã | AnimationNum, Frame | None |
| Get Animation Frame | 196 | C4 | Ä | AnimationNum | Frame |
| Stop All Animations | 198 | C6 | Æ | None | None |
| Clear Animation | 199 | C7 | Ç | AnimationNum | None |
| Clear All Animations | 200 | C8 | È | None | None |
| Resume All Animations | 201 | C9 | É | None | None |

*Table 43: Graphs Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| List All Bargraphs | 102 | 66 | f | None | BarType, BarValue |
| Define A Plain Bargraph | 103 | 67 | g | Index, Min, Max, X, Y, Width, Height, FGR, FGG, FGB, BGR, BGG, BGB, D | None |
| Define A 9-Slice Bargraph | 104 | 68 | h | Index, Min, Max, X, Y, Width, Height, BFG, BBG, D | None |
| Update A Bargraph Value | 105 | 69 | i | Index, Value | Result |
| Update Multiple Bargraph Values | 106 | 6A | j | Index, Length, Values | Result |
| Clear All Bargraphs | 107 | 6B | k | None | None |
| Reset a Trace Value | 109 | 6D | m | Index | None |
| Reset Multiple Trace Values | 110 | 6E | n | Index, Number | None |
| List All Traces | 115 | 73 | s | None | Index, Value |
| Initialize a Trace | 116 | 74 | t | Index, X, Y, Width, Height, Min, Max, Step, Style, Red, Green, Blue | None |
| Update a Trace | 117 | 75 | u | Index, Value | None |
| Update Multiple Traces | 118 | 76 | v | Index, Length, Values | Result |
| Clear All Traces | 119 | 77 | w | None | None |
| Set Trace Min/Max | 148 | 94 | ö | Index, Min, Max | None |
| Get Trace Min/Max | 149 | 95 | ò | Index | Min, Max |

*Table 44: Keypad Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Clear Key Buffer | 69 | 45 | E | None | None |
| Set Keypad Transmit Mode | 79 | 4F | O | AutoTransmit | None |
| Set Debounce Time | 85 | 55 | U | Mode | None |
| Set Typematic Interval | 158 | 9E | ž | Interval | None |
| Set Typematic Delay | 159 | 9F | Ÿ | Delay | None |
| Set Auto Repeat Mode | 165 | A5 | ¥ | Mode | None |
| Assign Keypad Codes | 213 | D5 | Õ | Length, KeyCodes | None |

*Table 45: Touch Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Create A Touch Region | 132 | 84 | „ | Index, X, Y, Width, Height, Up, Down | None |
| Clear A Touch Region | 133 | 85 | … | Index | None |
| Clear All Touch Regions | 134 | 86 | † | None | None |
| Change Touch Reporting Style | 135 | 87 | ‡ | ReportingType | None |
| Set Dragging Threshold | 137 | 89 | ‰ | Threshold | None |
| Calibrate Touch Screen | 139 | 8B | ‹ | None | Result |
| Load Region File | 140 | 8C | Œ | FileName | Result |
| Restore Touch Calibration | 141 | 8D | ⍰ | None | Result |
| Set Out of Region Setting | 142 | 8E | Ž | Setting | None |
| Get Out of Region Setting | 143 | 8F | ⍰ | None | Report |
| Set Region Activation State | 146 | 92 | Æ | Index, State | None |
| Get Region Activation State | 147 | 93 | ô | Index | State |

*Table 46: Output Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Set GPO State | 73 | 49 | I | Number, Setting | None |
| Activate Motor | 160 | A0 |  | Frequency, Duration | None |
| Set Input Feedback | 182 | B6 | ¶ | InputOutputType, DownFrequency, UpFrequency | None |
| Activate Buzzer and Motor | 183 | B7 | · | Frequency, Duration | None |
| Activate Buzzer | 187 | BB | » | Frequency, Duration | None |
| Set Default Buzzer Beep | 188 | BC | ¼ | Frequency, Duration | None |

*Table 47: Scripts Commands*

| Name | Dec | Hex | ASCII | Parameters | Response |
|---|---|---|---|---|---|
| Run Script File | 93 | 5D | ] | FileName | None |
| Create a Scripted Region | 131 | 83 | ƒ | ID, X, Y, W, H, UpBitmap, DownBitmap, UpScript, DownScript | None |
| Create a Scripted Key | 138 | 8A | Š | ID, Row, Col, UpScript, DownScript | None |

## 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 48: Data Types with Representations*

| | ANSI C/C++ | C# | Visual Basic |
|---|---|---|---|
| Byte | unsigned char | byte | Byte |
| Signed Byte | signed char | sbyte | SByte |
| Short | unsigned short | ushort | UShort |
| Signed Short | short | short | Short |
| Integer | unsigned int | uint | UInteger |
| Signed Integer | int | int | Integer |
| String | string | string | String |

*Table 49: Data Type Descriptions*

| | |
|---|---|
| **Byte** | Unsigned 8 bit data type that can represent a value from 0 to 255. |
| **Signed Byte** | Signed 8 bit data type that can represent a value from -128 to 127. |
| **Short*** | Unsigned 16 bit data type can represent values from 0 to 65,536. |
| **Signed Short*** | Signed 16 bit data type that can represent values from -32,768 to 32,767. |
| **Integer *** | Unsigned 32 bit data type that can represent values from 0 to 4,294,967,295. |
| **Signed Integer*** | Signed 32 bit data type that can represent values of -2,147,483,648 to 2,147,483. |
| **String** | Strings are a multiple character bytes terminated by a single null byte. The ASCII character set is used by default, but Unicode or UTF-8 strings may be used where specifically outlined. |

***Note:** Transmission of multiple byte values follows big endian order.

## 3.3 Colour Codes

*Table 50: 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 |

# 4 Definitions

9-Slice: Graphic format used to scale bitmaps, usually rectangular, without distorting their geometry. Nine regions define the object center, four corners, and four sides for accurate up or down scaling.

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.

GUI: Graphical user interface.

Hexadecimal: A base 16 number system utilizing symbols 0 through F to represent the values 0-15.

$I^2C$: 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, used in reference to a crisp, full-colour LCD technology.

USB: Universal Serial Bus protocol widely used in PCs.

# 5 Contact

### Sales

Phone: 403.229.2737
Email: sales@matrixorbital.ca

### Support

Phone: 403.229.2737
Email: support@matrixorbital.ca

### Design

Phone: 403.229.2737
Email: design@matrixorbital.ca

### Online

Purchase: www.matrixorbital.com
Support: www.matrixorbital.ca