



GLC24064
Technical Manual

Revision: 3.1

Contents

Contents	ii
1 Getting Started	1
1.1 Display Options Available	1
1.2 Accessories	1
1.3 Features	2
1.4 Connecting to a PC	3
1.5 Installing the Software	4
1.5.1 MOGD#	4
2 Hardware Information	5
2.1 DB-9 Connector	5
2.1.1 Power Through DB-9 Jumper	6
2.2 Power/Data Connector	6
2.2.1 Legacy Data Connector Jumpers	7
2.3 Protocol Select Jumpers	7
2.4 General Purpose Output	8
2.5 Manual Override	9
2.6 File System Lock Jumper	9
3 Troubleshooting	10
3.1 The display does not turn on when power is applied.	10
3.2 The display module is not communicating.	10
3.3 The display module is communicating, however text cannot be displayed.	11
3.4 There is a problem uploading fonts or bitmaps.	11
4 Communications	11
4.1 Introduction	11
4.1.1 I ² C Communication Summary	11
4.1.2 I ² C Transaction Example	12
4.1.3 Serial Communication	13
4.2 Turn Flow Control On	13
4.3 Turn Flow Control Off	14
4.4 Changing the I ² C Slave Address	14
4.5 Changing the Baud Rate	15
4.6 Setting a Non-Standard Baud Rate	16
5 Fonts	17
5.1 Introduction	17
5.1.1 Font File Format	18
5.1.2 Creating a Font	18
5.2 Uploading a Font File	20
5.3 Setting the Current Font	20
5.4 Font Metrics	21

5.5	Set Box Space Mode	21
6	Text	22
6.1	Introduction	22
6.1.1	Character Set	22
6.1.2	Control Characters	22
6.2	Move Cursor Home	22
6.3	Setting the Cursor Position	23
6.4	Setting the Cursor Coordinate	23
6.5	Auto Scroll On	23
6.6	Auto Scroll Off	24
7	Bitmaps	24
7.1	Introduction	24
7.2	Uploading a Bitmap File	24
7.3	Drawing a Bitmap from Memory	25
7.4	Drawing a Bitmap Directly	25
8	Bar Graphs and Drawing	26
8.1	Introduction	26
8.2	Set Drawing Color	26
8.3	Draw Pixel	27
8.4	Drawing a Line	27
8.5	Continue a Line	27
8.6	Draw a Rectangle	28
8.7	Drawing a Solid Rectangle	28
8.8	Initializing a Bar Graph	29
8.9	Drawing a Bar Graph	29
8.10	Initializing a Strip Chart	30
8.11	Shifting a Strip Chart	31
9	General Purpose Output	32
9.1	Introduction	32
9.2	General Purpose Output Off	32
9.3	General Purpose Output On	32
9.4	Set Startup GPO state	33
10	Display Functions	33
10.1	Introduction	33
10.2	Clear Screen	33
10.3	Display On	34
10.4	Display Off	34
10.5	Set Brightness	34
10.6	Set and Save Brightness	35
10.7	Set Contrast	35
10.8	Set and Save Contrast	36

11	Filesystem	36
11.1	Introduction	36
11.1.1	File Upload Protocol	36
11.1.2	XModem Upload Protocol	38
11.2	Wipe Filesystem	38
11.3	Deleting a File	40
11.4	Get Filesystem Space	40
11.5	Get Filesystem Directory	41
11.6	Filesystem Upload	41
11.7	Downloading a File	41
11.8	Moving a File	42
12	Data Security	42
12.1	Introduction	42
12.2	Set Remember	43
12.3	Data Lock	43
12.4	Set and Save Data Lock	45
12.5	Dump the Filesystem	45
12.6	Write Customer Data	45
12.7	Read Customer Data	45
13	Miscellaneous	46
13.1	Introduction	46
13.2	Read Version Number	46
13.3	Read Module Type	46
14	Command Summary	48
14.1	Communications	48
14.2	Fonts	48
14.3	Text	48
14.4	Bitmaps	49
14.5	Bar Graphs and Drawing	49
14.6	General Purpose Output	50
14.7	Display Functions	50
14.8	Filesystem	51
14.9	Data Security	51
14.10	Miscellaneous	52
14.11	Command By Number	52
15	Appendix	54
15.1	Specifications	54
15.1.1	Environmental	54
15.1.2	Electrical	54
15.2	Optical Characteristics	55
15.3	Physical Layout	55
15.4	Ordering Information	57
15.5	Definitions	57

15.6 Contacting Matrix Orbital 57
15.7 Revision History 58

1 Getting Started



Figure 1: GLC24064-WB

The GLC24064 is an intelligent graphic LCD display designed to decrease development time by providing an instant solution to any project. With the ability to communicate via serial RS-232/TTL and I²C protocols, the versatile GLC24064 can be used with virtually any controller. The ease of use is further enhanced by an intuitive command structure to allow display settings such as backlighting, contrast and baud rate to be software controlled. Additionally, text and fonts may be uploaded to the display and stored in the onboard memory.

1.1 Display Options Available

The GLC24064 comes in a variety of colors including the standard yellow/green, the popular blue/white and the crisp white/grey. Extended voltage, and temperature options are also available, to allow you to select the display which will best fit your project needs.

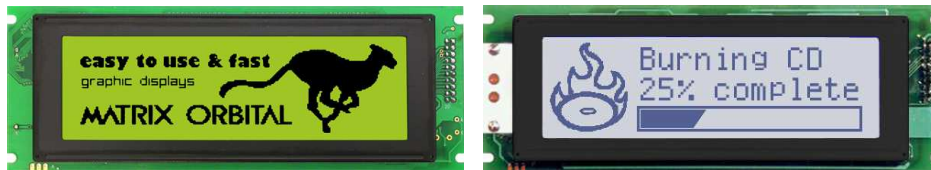


Figure 2: GLC24064 Options

1.2 Accessories

NOTE Matrix Orbital provides all the interface accessories needed to get your display up and running. You will find these accessories and others on our e-commerce website at <http://www.matrixorbital.com>. To contact a sales associate see Section 15.6 on page 57 for contact information.



Figure 3: 5V Power Cable Adapter



Figure 4: 12V Power Cable Adapter (V/VPT Models)



Figure 5: Breadboard Cable

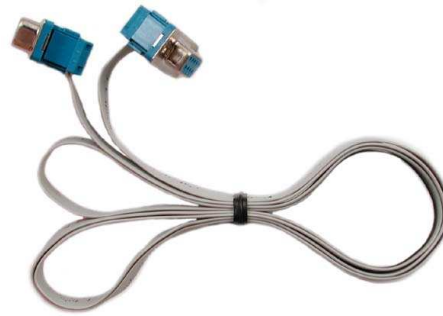


Figure 6: Serial Cable 4FT

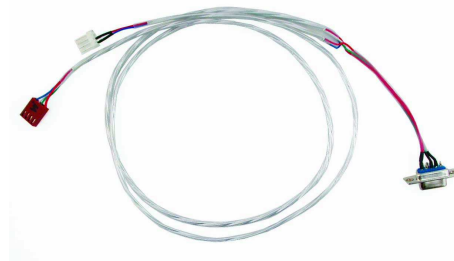


Figure 7: Communication and 5V Power Cable

1.3 Features

- 240 x 64 pixel graphics display

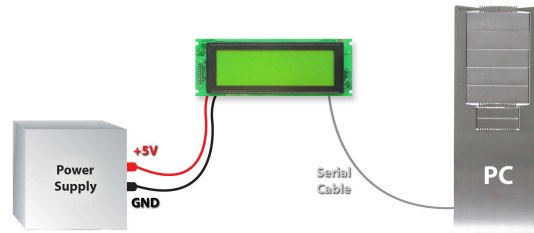
- Selectable communication protocol, Serial at RS-232 or TTL Levels or I²C
- 5V - 20mA General Purpose Output
- 16 KB flash memory for fonts and bitmaps
- Lightning fast communication speeds, up to 115 kbps for RS-232 and 100 kbps for I²C
- Adjustable contrast and backlight brightness
- Extended temperature available for extreme environments of -20C to 70C
- Extended voltage and efficient power supply available

1.4 Connecting to a PC

The GLC24064 connects seamlessly to a PC and it is an excellent means of testing the functionality and uploading new fonts and bitmaps. You will require a standard RS-232 9-pin serial cable such as the one pictured in *figure 6 on the previous page*, as well as a modified 5V power adapter such as the one pictured in *figure 3 on the preceding page*.

In order to connect your display to a personal computer follow these easy instructions:

1. Plug the serial cable into the com port you wish to use.
2. Connect the modified 5V power adapter to a power lead from your PC power supply (you will have to open your computer case).
3. Connect the serial cable to the DB-9 connector on the back of the display.
4. Connect the 5V power adapter to the 4-pin connector on the back of the display.



WARNING DO NOT use the standard floppy drive power connector, as this will not provide you with the correct voltage and will damage the display module.

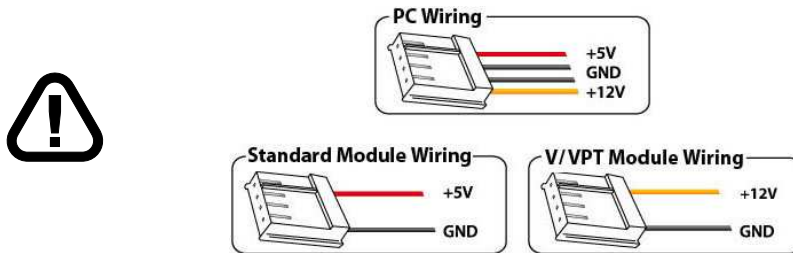


Figure 8: PC vs Matrix Orbital Display Module Wiring

1.5 Installing the Software

1.5.1 MOGD#

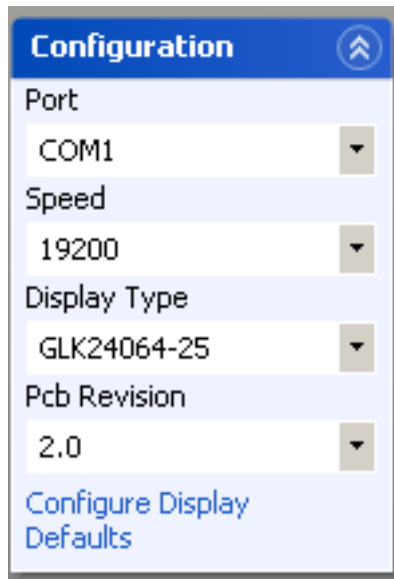
MOGD# is the latest updated version of MOGD and can be used to manage font and graphics downloads as well as exercise all of the features of our graphical displays. MOGD# provides a new user friendly interface as well as many feature enhancements.

To install MOGD# from the Matrix Orbital CD, follow the following steps:

1. Insert the Matrix Orbital Cd-ROM into the Cd drive
2. Locate the file *MogdSharp.zip*, which should be in the “CD-drive:\Download” directory
3. Unzip *MogdSharp.zip* to a temporary directory using a program such as Winzip, Pkzip, etc.
4. Double click on "setup.exe"
5. Follow the instructions on the screen to complete the installation
6. MOGD# requires the .NET framework 2.0 and will download and install it automatically

After the installation is complete there will be a Matrix Orbital entry under “Start->Programs->Matrix Orbital” in the start menu. Click on the 'Mogd Sharp' entry to run the program.

Be sure to check the information selected in the configuration panel the first time MOGD# is run. Once this information is entered correctly the program can be used to control all functions of the graphic display.



Port	The serial port the display is plugged in to.
Speed	The communication speed the display module is set to. (Default 19,200)
Display Type	The type of display. (GLC24064)
PCB Revision	The revision of the display you are using. (Found on the back of the PCB)

Figure 9: Mogd Sharp Settings

NOTES

- MOGD# may also be downloaded from Matrix Orbital's support site at http://www.matrixorbital.ca/software/software_graphic/MogdSharp
 - Winzip is available as a free download from <http://www.winzip.com>
-

2 Hardware Information

Refer to the following diagram for this chapter:

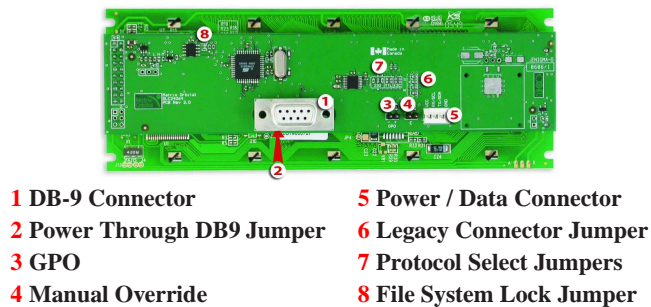
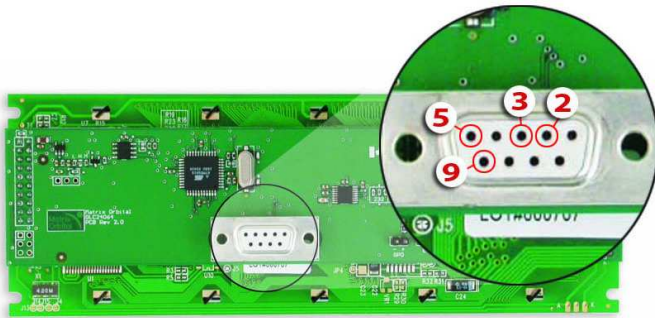


Figure 10: GLC24064

2.1 DB-9 Connector

The GLC24064 provides a *DB-9 Connector* to readily interface with serial devices which use the EIA232 standard signal levels of $\pm 12V$ to $\pm 12V$. It is also possible to communicate at TTL levels of 0 to +5V by setting the *Protocol Select Jumpers* to TTL. As an added feature it is also possible to apply power through pin 9 of the *DB-9 Connector* in order to reduce cable clutter. However, in order to accomplish this you must set the *Power Through DB-9 Jumper*.



- Pin 2** Rx \ SCL (I²C clock)
- Pin 3** Tx \ SDA (I²C data)
- Pin 5** GND
- Pin 9** PWR (Must solder Power Through DB-9 Jumper. See table ?? on page ?? for power requirements.)

Figure 11: RS-232 Pin out

2.1.1 Power Through DB-9 Jumper

In order to provide power through pin 9 of the *DB-9 Connector* you must place a solder jumper on the *Power through DB-9 Jumper* pictured in *figure 12* below. The GLC24064 allows all voltage models to use the power through DB-9 option, see table ?? on page ?? for display module voltage requirements.



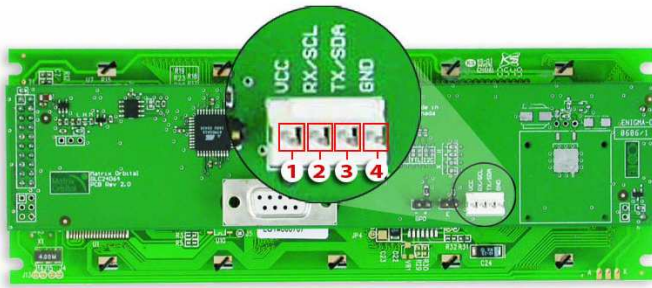
Figure 12: Power Through DB-9 Jumper



WARNING Do not apply voltage through pin 9 of the DB-9 connector AND through the Power/Data Connector at the same time.

2.2 Power/Data Connector

The *Power/Data Connector* provides a standard connector for powering the display module. The GLC24064 requires five volts for the standard display module, between nine to fifteen for the wide voltage (V) and between nine to thirty-five volts for the wide voltage with efficient power supply module (VPT). The voltage is applied through pins one and four of the four pin *Power/Data connector*. Pins two and three are reserved for serial transmission, using either the RS-232/TTL or the I²C protocol, depending on what has been selected by the *Protocol Select Jumpers*. Pins two and three may be reversed by changing the *Legacy Connector Jumpers* in order to be compatible with previous PCB revisions.



- Pin 1** PWR (See table ?? on page ??)
- Pin 2** Rx \ SCL (I²C clock)
- Pin 3** Tx \ SDA (I²C data)
- Pin 4** GND

Figure 13: Power Connector and Pin out

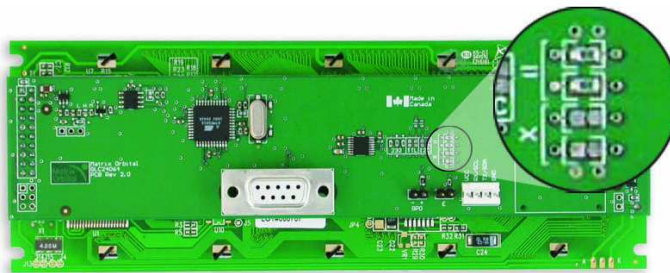


WARNINGS

- Do not apply any power with reversed polarization.
- Do not apply any voltage other than the specified voltage.

2.2.1 Legacy Data Connector Jumpers

To reverse pins two and three of the *Power/Data Connector* remove the zero ohm resistors from the *Legacy Data Connector Jumpers*, labeled with the = symbol and place them on the jumpers labeled with the X symbol. This will allow you to transmit on pin two, and receive data on pin three instead of the default of receiving on pin two and transmitting on pin three of the *Power/Data Connector*.



Power/Data Connector	
Pin 2	Rx \ SCL (I ² C clock)
Pin 3	Tx \ SDA (I ² C data)
Pin 2	Tx \ SDA (I ² C data)
Pin 3	Rx \ SCL (I ² C clock)

Figure 14: Legacy Data Connector Jumpers

2.3 Protocol Select Jumpers

The *Protocol Select Jumpers*, pictured below in *figure 15*, provide the means necessary to toggle the display module between RS-232, TTL and I²C protocols. As a default, the jumpers are set to RS-232 mode with zero ohm resistors on the 232 jumpers. In order to place the display module in I²C mode you must first remove the zero ohm resistors from the 232 jumpers and then solder the resistors on to the I²C jumpers.

The display will now be in I²C mode and have a default slave address of 0x50 unless it has been changed. Similarly, in order to change the display to TTL mode, simply remove the zero ohm resistors from the 232 or I²C jumpers and solder them to the TTL jumpers.

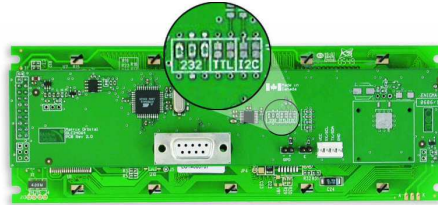


Figure 15: Protocol Select Jumpers

2.4 General Purpose Output

A unique feature of the GLC24064 is the ability to control relays and other external devices using a *General Purpose Output (3)*, which can provide up to 20 mA of current and +5Vdc from the positive side of the GPO. This is limited by a 240 ohm resistor which is located directly above the positive pin as pictured below in *figure 16*. If the device, which is being driven by a GPO, requires a relatively high current (such as a relay) and has an internal resistance of its own greater than 250 ohms, then the 240 ohm resistor may be removed and replaced with a Jumper.



Pin 1 - GND
Pin 2 + MAX: 20 mA, +5Vdc

Figure 16: General Purpose Output



WARNING If connecting a relay, be sure that it is fully clamped using a diode and capacitor in order to absorb any electro-motive force (EMF) which will be generated.

2.5 Manual Override

The *Manual Override* is provided to allow the GLC24064 to be reset to factory defaults. This can be particularly helpful if the display module has been set to an unknown baud rate or I²C Slave Address and you are no longer able to communicate with it. If you wish to return the module to its default settings you must:

1. Power off the display module.
2. Place a Jumper on the *Manual Override* pins.
3. Power up the display module.
4. The display module is now set to its default values listed below in *table 1*.
5. Edit and save settings.

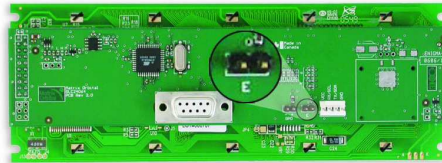


Figure 17: Manual Override Jumper

Table 1: Default Values

Contrast	128
Backlight	255
Baud Rate	19.2 kbps
I²C Slave Address	0x50
Data Lock	False
RS232AutoTransmitData	True

NOTE The display module will revert back to the old settings once turned off, unless the settings are saved.

2.6 File System Lock Jumper

The *File System Lock Jumper* allows you to lock the file system on the GLC24064 so that no fonts or bitmaps can be either written or deleted from the on board memory. This feature is useful in order to protect data integrity of production units, if protection of other settings is required see *Chapter 12 Data Security*.

To lock the file system, solder a zero ohm resistor or use a solder jumper on the *Filesystem Lock Jumper* pictured in *figure 18* below.

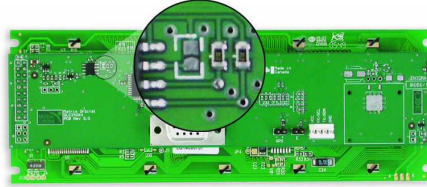


Figure 18: File System Lock Jumper

3 Troubleshooting

3.1 The display does not turn on when power is applied.

- First, you will want to make sure that you are using the correct power connector. Standard floppy drive power cables from your PC power supply may fit on the Power/Data Connector however they do not have the correct pin out as can be seen in *figure 8 on page 3*. Matrix Orbital supplies power cable adapters for connecting to a PC, which can be found in the *Accessories Section on page 1*.
- The next step is to check the power cable which you are using for continuity. If you don't have an ohm meter, try using a different power cable, if this does not help try using a different power supply.
- The last step will be to check the *Power / Data Connector* on the GLC24064. If the *Power / Data Connector* has become loose, or you are unable to resolve the issue, please contact Matrix Orbital see *15.6 on page 57* for contact information.

3.2 The display module is not communicating.

- First, check the communication cable for continuity. If you don't have an ohm meter, try using a different communication cable. If you are using a PC try using a different Com port.
- Second, please ensure that the display module is set to communicate on the protocol that you are using, by checking the *Protocol Select Jumpers*. To change the protocol used by the display module see *Section 2.3 on page 7*.
- Third, ensure that the host system and display module are both communicating on the same baud rate. The default baud rate for the display module is 19200 bps.
- If you are communicating to the display via I²C please ensure that the data is being sent to the correct address. The default slave address for the display module is 0x50.

NOTE I²C communication will always require pull up resistors.

- Finally, you may reset the display to its default settings using the *Manual Override Jumper*, see *Section 2.5 on page 9*.

3.3 The display module is communicating, however text cannot be displayed.

- The cause of this is often that no font has been loaded onto the display. To load a font onto the display see *Section 4.2.1 on page 16*.
- Another common cause may be that the contrast settings have been set to low. The solution to this problem is to adjust the contrast settings, the default setting that will work in most environments is 128.

NOTE Optimal contrast settings may vary according to factors such as temperature, viewing angle and lighting conditions.

3.4 There is a problem uploading fonts or bitmaps.

- First, ensure that you can communicate to the display. A good test is to use a PC, with MOGD# installed, to connect to the display. See *Section 1.4 on page 3* for setting up a PC to test the GLC24064.
- Second, ensure that the *File System Lock Jumper* has not been set. See *Section 2.6 on page 9*.
- Third, please ensure that the display module's memory is not full. The GLC24064 has 16 Kb of memory for fonts and bitmaps.

NOTE If you are unable to resolve any issue please contact Matrix Orbital. See *15.6 on page 57* for contact information.

4 Communications

4.1 Introduction

The commands listed in this chapter describe how to configure data flow on the .

4.1.1 I²C Communication Summary

The is capable of communicating at 100 KHz in I²C mode, with 127 units addressable on a single I²C communication line. However, in order to communicate via I²C you must first ensure that pull up resistors, with a nominal value of 1K to 10K, are placed on the SCL and SDA communication lines coming from pins two and three of the Data / Power Connector respectively. Data responses by the module are automatically output via RS232, in case the host will be querying the module, it is necessary for the host to inform the module that its responses are to be output via I²C. This can be done by sending command 254 /160 / 0 to

turn off auto transmission of data in RS232. This will keep the data in the buffer until the master clocks a read of the slave. The I²C data lines operate at 5V normally or 3.3V for -1U style units. The uses 8-bit addressing, with the 8th or Least Significant Bit (LSB) bit designated as the read/write bit, a 0 designates a write address and a 1 designates a read address. The default read address of the display module will be 0x51, whereas the write address is 0x50 by default. This address may be changed by using cmd 254 / 51 / <address>. The should only be sent addresses that are even (LSB is 0). When the I²C master wishes to write to the display, the effective address is \$50 (0101 0000) , since the LSB has to be 0 for an I²C master write. When the I²C master wishes to read the , the effective address is \$51 (0101 0001), since the LSB has to be 1 for an I²C master read.

If we take a standard Phillips 7 bit address of \$45 (100 0101), Matrix Orbital's would describe this Phillips I²C address as \$8A (1000 1010). The read address would be \$8B (1000 1011).

The unit does not respond to general call address (\$00).

When communicating in I²C the will send an ACK on the 9th clock cycle when addressed. When writing to the display module, the display will respond with a ACK when the write has successfully been completed. However if the buffer has been filled, or the module is too busy processing data it will respond with a NAK. When performing a multiple byte read within one I²C transaction, each byte read from the slave should be followed by an ACK to indicate that the master still needs data, and a NAK to indicate that the transmission is over.

The has some speed limitations, especially when run in I²C mode. Here are some considerations when writing I²C code:

- * to be able to read the replies of query commands (eg. cmds 54, 55) the following command must be sent (only needs to be sent once, so this can be done somewhere in init): 254 / 160 / 0 this command puts the reply data in the I²C output buffer instead of the RS232 output buffer. Please note that due to a 16 byte output buffer, query commands that reply with more than 16 bytes cannot be read (eg cmd Get FileSystem Directory)

- * 3ms delay between the read commands

- * 625us delay in between data bytes within a transaction is necessary

- * 375us between transactions is necessary

NOTE These delays are consrevative, and may be decreased based on performance

4.1.2 I²C Transaction Example

The typical I²C transaction contains four parts: the start sequence, addressing, information, and stop sequence. To begin a transaction the data line, SDA, must toggle from high to low while the clock line, SCL, is high. Next, the display must be addressed using a one byte hexadecimal value, the default to write to the unit is 0x50, while read is 0x51. Then information can be sent to the unit; even when reading, a command must first be sent to let the unit know what type of information it is required to return. After each bit is sent, the display will issue an ACK or NACK as described above. Finally, when communication is complete, the transaction is ended by toggling the data line from low to high while the clock line is high. An example of the use of this algorithm to write a simple "HELLO" message can be seen in 2.

Table 2: I²C Transaction Algorithm

START	Toggle SDA high to low
Address	0x50
Information	0x48 0x45 0x4C 0x4C 0x4F
STOP	Toggle SDA low to high

4.1.3 Serial Communication

In addition to being able to communicate via I²C the communicates natively through the RS-232 protocol at a default baud rate of 19,200 bps and is capable of standard baud rates from 9600 to 115,200 bps. Furthermore the is also capable of reproducing any non-standard baud rate in between using values entered into our baud rate generation algorithm and set through command 164 (0xA4). The display module communicates at standard voltage levels of -30V to +30V or at TTL levels of 0 to +5V by setting the *Protocol Select Jumpers* to TTL.

4.2 Turn Flow Control On

Syntax	Hexadecimal	0xFE 0x3A [full] [empty]	
	Decimal	254 58 [full] [empty]	
	ASCII	254 “:” [full] [empty]	
Parameters	Parameter	Length	Description
	full	1	Bytes remaining before issuing a almost full message. (Full is 0)
	empty	1	Bytes available before issuing a almost empty message. (Empty is 128)

Description This command enables flow control. When the buffer fills so that only [full] bytes are available, the display will return an “almost full” message (0xFE) to the host controller. When the buffer empties so that only [empty] bytes remain, the display will return an “almost empty” message (0xFF) to the host controller.

The display will return the “almost full” message for every byte sent to the display until the used buffer space once more drops below the [full] level. Whether the user is in ‘flow control mode’ or not, the module will ignore display or command bytes which would overrun the buffer.

While in ‘flow control mode’ the unit will return 0xFE when buffer is almost full even though it may have already thrown rejected data away. The buffer size for the display is 128 bytes.

When using this command in an application, selection of the value for the buffer [full] should be considered very carefully. This is a critical aspect to be able to use this feature to it’s full potential. When using a host system or PC which contains a FIFO, the user should set the value of [full] equal to or greater than the size of the FIFO. The reason for this is that the FIFO may be full when the host system receives 0xFE. In the case of 16550 UART the size at its maximum is 16, therefore the value of [full] should be set to 16 or greater. It is suggested that the “almost full” parameter be equal to the largest chunk of data the host will be sending the display (should be less than 127).

NOTE This command is not available in I²C mode.

Remembered Yes
 Default Off

4.3 Turn Flow Control Off

Syntax Hexadecimal 0xFE 0x3B
 Decimal 254 59
 ASCII 254 “;”

Description This command turns off flow control. Bytes may overflow the buffer without warning.

NOTE This command is not available in I²C mode.

Remembered Yes

4.4 Changing the I²C Slave Address

Syntax	Hexadecimal	0xFE 0x33 [adr]	
	Decimal	254 51 [adr]	
	ASCII	254 “3” [adr]	
Parameters	Parameter	Length	Description
	adr	1	The new I ² C write address (0x00 - 0xFF).
Description	This command sets the I ² C write address of the module between 0x00 and 0xFF. The I ² C write address must be an even number and the read address is automatically set to one higher. For example if the I ² C write address is set to 0x50, then the read address is 0x51.		

NOTE The change in address is immediate.

Remembered	Always
Default	0x50

4.5 Changing the Baud Rate

Syntax	Hexadecimal	0xFE 0x39 [speed]	
	Decimal	254 57 [speed]	
	ASCII	254 “9” [speed]	
Parameters	Parameter	Length	Description
	speed	1	Hex value corresponding to a baud rate.

Description This command sets the RS-232 port to the specified [speed]. The change takes place immediately. [speed] is a single byte specifying the desired port speed. Valid speeds are shown in the table below. The display module can be manually reset to 19,200 baud in the event of an error during transmission, including transmitting a value not listed below, by setting the manual override jumper during power up. However, it should be noted that this command will be ignored until the manual override jumper is removed again.

Hex Value	Baud Rate
0xCF	9600
0x8A	14400
0x67	19200
0x44	28800
0x33	38400
0x22	57600
0x19	76800
0x10	115200

NOTE This command is not available in I²C mode.

Remembered Always
 Default 19,200 bps

4.6 Setting a Non-Standard Baud Rate

Syntax Hexadecimal 0xFE 0xA4 [speed]
 Decimal 254 164 [speed]

Parameter	Length	Description
speed	2	Inputed LSB MSB from baud rate formula (12-2047).

Description This command sets the RS-232 port to a non-standard baud rate. The command accepts a two byte parameter that goes directly into the modules baud generator. Use the formula, $speed = \frac{CrystalSpeed}{8 \times DesiredBaud} - 1$ to calculate the [speed] for any baud rate setting. The speed can be anywhere from 12 to 2047 which corresponds to a baud range of 977 to 153,800 baud. Setting the baud rate out of this range could cause the display to stop working properly and require the Manual Override jumper to be set.

Remembered Always

Examples

Crystal Speed 16 Mhz

Desired BAUD 13,500

$$speed = \frac{crystalspeed}{8 * DesiredBaud} - 1 \quad speed = \frac{16,000,000}{8 * 13,500} - 1$$

$$speed = 148.15 - 1$$

$$speed = 147.15$$

- **LSB** = 0x93 (rounded)
- **MSB** = 0x00
- Intended Baud Rate: 13,500 baud Actual Baud Rate:
 $\frac{16,000,000}{8(147+1)} = 13,514$ Percent Difference: 0.1%

NOTES

- Results from the formula are rounded down to the nearest whole number (i.e 73.07 = 73).
- This formula becomes less accurate as baud rates increase, due to rounding.
- Place the speed result backwards into the formula to receive the actual baud rate.
($Baud = \frac{CrystalSpeed}{8(speed+1)}$)
- The actual baud rate must be within 3% of the intended baud rate for the device to communicate.

NOTES

- This command is not available in I²C mode.
-

5 Fonts

5.1 Introduction

The comes loaded with the 'Small Filled' and 'Futura Bk BT 16' fonts by default. However, it is capable of displaying any font that is uploaded to it in the correct format. MOGD# provides a simple method of generating font files from your installed fonts. For instructions on how to install MOGD# see *Section 1.5.1 on page 4*.

5.1.1 Font File Format

A font file consists of three parts, a header, a character table and bitmap data.

1. Header (4 bytes)
 - (a) Nominal Width (1 byte)
 - (b) Height (1 byte)
 - (c) ASCII Start Value (1 byte)
 - (d) ASCII End Value (1 byte)
2. Character Table (3 bytes for every character between the ASCII Start and End values inclusive)
 - (a) High Offset MSB (1 byte)
 - (b) Low Offset LSB(1 byte)
 - (c) Character Width (1 byte)
3. Bitmap Data

5.1.2 Creating a Font

The following is an example of how to create a font file for the letters *h*, *i* and *j*.

First you must create the bitmaps containing the character data in bitmap form. *Figure 19* below illustrates the bit pattern for the *h*, *i* and *j* bitmap data.

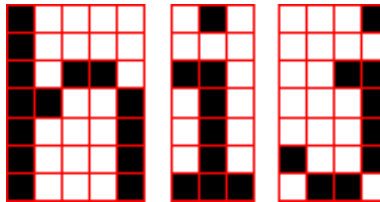


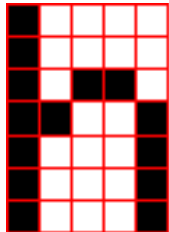
Figure 19: Bitmaps for h, i, and j

Second you may begin to create the font file starting with the header. The header will contain the nominal width, the height and the ASCII start and end values inclusive that you wish to create characters for.

Table 8: Font File Header

Nominal Width	Height	ASCII Start Val	ASCII End Val
0x05	0x07	0x68	0x6A

Next we will have to find out how many bytes each character will use up, in order to create the character table. The bitmaps are encoded horizontally and may have variable widths, *h* has a width of five, *i* a width of three and *j* a width of four, see the figure below for an example of encoding the first letter *h*:



Bitmap Data					Byte	Hex Value
1	0	0	0	0		
1	0	0	0	0	10000100	0x84
1	0	1	1	0	00101101	0x2D
1	1	0	0	1	10011000	0x98
1	0	0	0	1	11000110	0xC6
1	0	0	0	1	00100000	0x20
1	0	0	0	1		

Figure 20: Bitmap Encoding

As you can see the letter *h* will take up five bytes with the last five bits being zero padded to form a full byte. So if you continue the process you will get the character data as seen in *table 5.1.2*.

Character Data

	Character Data					Byte Size (For Reference)
<i>h</i>	0x84	0x2D	0x98	0xC6	0x20	0x05
<i>i</i>	0x43	0x24	0x84			0x03
<i>j</i>	0x2D	0x98	0x19	0x60		0x04

The second part of the font file is the character table. The character table is comprised of three bytes for every glyph in the font file.

The first two bytes represents the position, in bytes, of the glyph stored MSB LSB referenced from the beginning of the file (including the header. The third byte is the width of the glyph in pixels. So because there will be 0x09 bytes in the character table (three bytes for each glyph) and four bytes in the header section, the first entry in the table will be 13, or 0x00 0x0D in hexadecimal, and 0x05 for the width.

To calculate the second entry in the character table, representing the position and width of the second glyph, take the offset of the first entry and add the size of the first bitmap in bytes. Since the first glyph occupies 0x05 bytes as seen in table 5.1.2 above, and the offset is 0x00 0x0D, the offset of the second entry will be 0x00 0x12 and the width of the glyph is 0x03.

Calculate the third entry the same way as the second to get *table 9* below.

Table 9: Character Table

	High Offset (MSB)	Low Offset (LSB)	Character Width
<i>h</i>	0x00	0x0D	0x05
<i>i</i>	0x00	0x12	0x03
<i>j</i>	0x00	0x15	0x04

Once completed, place the character table after the header and the character data at the end, as seen in *table 10*.

Table 10: Sample Font File

0x05	0x07	0x68	0x6A	0x00	0x0D	0x05	0x00
0x12	0x03	0x00	0x15	0x04	0x84	0x2D	0x98
0xC6	0x20	0x43	0x24	0x84	0x2D	0x98	0x19
0x60							

Red = Header

Blue = Character Table

Purple = Character Data

5.2 Uploading a Font File

Syntax	Hexadecimal	0xFE 0x24 [refID] [size] [data]	
	Decimal	254 36 [refID] [size] [data]	
	ASCII	254 "\$" [refID] [size] [data]	
Parameters	Parameter	Length	Description
	refID	1	A unique font identification number.
	size	2	Font file size (LSB to MSB).
	data	x	Font file data.
Description	<p>In order to upload a font to the you must first initiate the upload font file command (0xFE 0x24), you must then pass it a reference identification number, which must be unique for every font on the display module. You may then pass the display module the two byte file size, which needs to be transferred LSB, then MSB. The last part of uploading a font is transmitting the font file data.</p> <p>For detailed instructions on uploading a file to the see <i>Section 11 on page 36</i>.</p>		

NOTE This command is available but not supported in I²C.

Remembered Always

5.3 Setting the Current Font

Syntax	Hexadecimal	0xFE 0x31 [refID]	
	Decimal	254 49 [refID]	
	ASCII	254 "1" [refID]	
Parameters	Parameter	Length	Description
	refID	1	A unique font identification number.

Description	In order to set the font on the you must know the font identification number of the font that you wish to use. The font ID is established when the font is saved to the display. The default installed fonts are “Small Filled” and “Futura Bk BT 16” and their font ID’s are 0x01 and 0x02 respectfully, with “Small Filled” being the default selected font. Once you are aware of the font ID for the font that you wish you use, simply send the command bytes (0xFE 0x31) and then send the font ID corresponding to the font. A directory listing of the contents of the entire filesystem may be obtained by using the “Get Filesystem Directory” command, see Section 11.5 on page 41 for more detailed information.
Remembered	Yes

5.4 Font Metrics

Syntax	Hexadecimal	0xFE 0x32 [lm] [tm] [csp] [lsp] [srow]	
	Decimal	254 50 [lm] [tm] [csp] [lsp] [srow]	
	ASCII	254 “2” [lm] [tm] [csp] [lsp] [srow]	
Parameters	Parameter	Length	Description
	lm	1	Left margin: Location in pixels.
	tm	1	Top margin: Location in pixels.
	csp	1	Character Spacing: Amount of space in pixels between characters.
	lsp	1	Line Spacing: Amount of space between lines in pixels.
	srow	1	Scroll Row: The Y location of the last row in pixels.
Description	Font metrics define where the characters are positioned on the screen, by setting where the rows and columns begin based on the [lm][tm][csp][lsp][srow] parameters. [lm] defines the leftmost position and [tm] the topmost. [csp] controls the amount of pixels that are placed in between characters and [lsp] controls the amount of pixels that are placed in between lines. [srow] is the location of the top of the last row that will be displayed on the . It defines the row that, when filled, will cause the display to auto scroll if auto scrolling is enabled. The font metrics will have to be reconfigured after changing to a different font.		
Remembered	Yes		

5.5 Set Box Space Mode

Syntax	Hexadecimal	0xFE 0xAC [value]
	Decimal	254 172 [value]

Parameters	Parameter	Length	Description
	value	1	Value (0: Off, 1: On)
Description	This command will toggle the box space mode. Box space mode is when a box, the size of the character to be written, is printed to the display before a character is written.		
Remembered	Yes		
Default	On		

6 Text

6.1 Introduction

The is an intelligent display module, designed to reduce the amount of code necessary to begin displaying data. This means that it is able to display all ASCII formatted characters and strings that are sent to it, which are defined in the current character set. The display module will begin displaying text at the top left corner of the display area, known as home, and continue to print to the display as if it was a page on a typewriter. When the text reaches the bottom right row, it is able to automatically scroll all of the lines up and continue to display text, with the auto scroll option set to on.

6.1.1 Character Set

The graphic displays such as the , do not have built in character sets. Instead fonts are uploaded to the display using the commands detailed in Section 5 on page 17.

6.1.2 Control Characters

In addition to a full text set, the display supports the following ASCII Control characters:

0x0A Line feed / New line - when this value is not defined in the font file. This command will create a new line on the display. If scrolling is on and the display is at the bottom of the screen, the whole screen is scrolled up.

6.2 Move Cursor Home

Syntax	Hexadecimal	0xFE 0x48
	Decimal	254 72
	ASCII	254 "H"
Description	This command moves the text insertion point to the top left of the display area (Row 1, Column 1).	
Remembered	No	

6.3 Setting the Cursor Position

Syntax	Hexadecimal	0xFE 0x47 [col] [row]									
	Decimal	254 71 [col] [row]									
	ASCII	254 “G” [col] [row]									
Parameters	<table border="1"><thead><tr><th>Parameter</th><th>Length</th><th>Description</th></tr></thead><tbody><tr><td>col</td><td>1</td><td>Column</td></tr><tr><td>row</td><td>1</td><td>Row</td></tr></tbody></table>	Parameter	Length	Description	col	1	Column	row	1	Row	
Parameter	Length	Description									
col	1	Column									
row	1	Row									
Description	This command sets the text insertion point to the [col] and [row] specified. The insertion point is positioned using the base size of the current font (this command does not position the insertion point at a specific pixel). The column used is determined by multiplying the width of the widest character in the font by the [column]. The row used is determined by multiplying the height of the font by [row + Metrics: line spacing].										
Remembered	No										

6.4 Setting the Cursor Coordinate

Syntax	Hexadecimal	0xFE 0x79 [x] [y]									
	Decimal	254 121 [x] [y]									
	ASCII	254 “y” [x] [y]									
Parameters	<table border="1"><thead><tr><th>Parameter</th><th>Length</th><th>Description</th></tr></thead><tbody><tr><td>x</td><td>1</td><td>The horizontal position in pixels.</td></tr><tr><td>y</td><td>1</td><td>The vertical position in pixels.</td></tr></tbody></table>	Parameter	Length	Description	x	1	The horizontal position in pixels.	y	1	The vertical position in pixels.	
Parameter	Length	Description									
x	1	The horizontal position in pixels.									
y	1	The vertical position in pixels.									
Description	This command positions the insertion point at a specific pixel (X,Y), which references the top left corner of the font insertion point.										
Remembered	No										

6.5 Auto Scroll On

Syntax	Hexadecimal	0xFE 0x51
	Decimal	254 81
	ASCII	254 “Q”
Description	When auto scrolling is on, it causes the display to shift the entire display’s contents up to make room for a new line of text when the text reaches the end of the scroll row defined in the font metrics (the bottom right character position) see <i>Section 5.4 on page 21</i> .	
Remembered	Yes	

Default On

6.6 Auto Scroll Off

Syntax Hexadecimal 0xFE 0x52
 Decimal 254 82
 ASCII 254 “R”

Description When auto scrolling is disabled, text will wrap to the top left corner of the display area when the text reaches the end of the scroll row defined in the font metrics (the bottom right character position) see *Section 5.4 on page 21*. Existing text in the display area is not erased before new text is placed. A series of spaces followed by a “Cursor Home” command may be used to erase the top line of text.

Remembered Yes

7 Bitmaps

7.1 Introduction

One of the main features of the is its ability to display bitmap images, that are either loaded onto its on board memory, or written directly to the screen. This chapter will cover creating a bitmap, uploading the bitmap, as well as drawing the bitmap from memory and directly.

7.2 Uploading a Bitmap File

Syntax Hexadecimal 0xFE 0x5E [refID] [size] [data]
 Decimal 254 94 [refID] [size] [data]
 ASCII 254 “^” [refID] [size] [data]

Parameters	Parameter	Length	Description
	refID	1	A unique bitmap identification number.
size	2	Bitmap file size (LSB to MSB).	
data	x	Bitmap data.	

Description The is capable of storing 128 font and bitmap files up to 16 Kbytes total. In order to upload a bitmap to the you must first initiate the upload font file command (0xFE 0x5E), you must then pass it a reference identification number, which must be unique for every font on the display module. You may then pass the display module the two byte system size, which needs to be transfered LSB, then MSB. This is almost always the entire 16kB, meaning the values 0x00 0x40 0x00 0x00 must be issued. The last part of uploading a bitmap is transmitting the bitmap file data.
For detailed instructions on uploading a file to the see *Section 11 on page 36*.

NOTE This command is available but not not supported in I²C.

Remembered Always

7.3 Drawing a Bitmap from Memory

Syntax	Hexadecimal	0xFE 0x62 [refID] [X] [Y]	
	Decimal	254 98 [refID] [X] [Y]	
	ASCII	254 “b” [refID] [X] [Y]	
Parameters	Parameter	Length	Description
	refID	1	The bitmap identification number.
	X	1	Left bounds.
	Y	1	Top bounds.
Description	This command will draw a bitmap that is located in the on board memory. The bitmap is referenced by the bitmaps reference identification number, which is established when the bitmap is uploaded to the display module. The bitmap will be drawn beginning at the top left, from the specified X,Y coordinates. A directory listing of the contents of the entire filesystem may be obtained by using the “Get Filesystem Directory” command, see Section 11.5 on page 41 for more detailed information.		
Remembered	No		

7.4 Drawing a Bitmap Directly

Syntax	Hexadecimal	0xFE 0x64 [X] [Y] [W] [H] [D]			
	Decimal	254 100 [X] [Y] [W] [H] [D]			
	ASCII	254 “d” [X] [Y] [W] [H] [D]			

Parameters	Parameter	Length	Description
	X	1	Left bounds.
	Y	1	Top bounds.
	W	1	Width
	H	1	Height
	D	(width*height)/8	Data

Description Drawing a bitmap to the , without first uploading the image to the memory can be a very useful feature for drawing images that are not used very often. In order to accomplish this, you must supply the display module with the X,Y coordinates, representing the top left corner of where you would like to draw the bitmap on the screen, as well as the width and the height of the bitmap. After you have supplied this data you may then upload the bitmap data to the . The length of this file is the bitmap width multiplied by height, divided by eight. The bitmap data is encoded into bytes horizontally and is transferred the same as if you were uploading a file, see *Section 11 on page 36* for more information about transferring data to the display module.

NOTE Drawing a bitmap directly to the display is supported by flow control. This command is available but not support in I²C mode.

Remembered No

8 Bar Graphs and Drawing

8.1 Introduction

Supplementary to the ability of the to display bitmaps and fonts, the also allows for a robust 2D drawing environment. With the ability to draw by pixel, line or rectangle, as well as the ability to continue a line to form a polygon, we are certain that you will spend less time, developing and creating better looking projects. With the addition of custom bar and strip graphs, you are sure to find the right tools to make any graphical layout a success.

8.2 Set Drawing Color

Syntax	Hexadecimal	0xFE 0x63 [color]	
	Decimal	254 99 [color]	
	ASCII	254 "c" [color]	
Parameters	Parameter	Length	Description
	color	1	Drawing color (0: White, 1-255: Black).

Description This command sets the drawing color for subsequent graphic commands that do not have the drawing color passed as a parameter. The parameter [color] is the value of the color where white is 0 and black is 1-255.

Remembered No

8.3 Draw Pixel

Syntax Hexadecimal 0xFE 0x70 [x] [y]
Decimal 254 112 [x] [y]
ASCII 254 "p" [x] [y]

Parameters	Parameter	Length	Description
	x	1	X screen location.
	y	1	Y screen location.

Description This command will draw a pixel at (x,y) using the current drawing color. The unit processes these requests fast enough to keep up with a steady stream at 115 Kbps so flow control is not required.

Remembered No

8.4 Drawing a Line

Syntax Hexadecimal 0xFE 0x6C [x1] [y1] [x2] [y2]
Decimal 254 108 [x1] [y1] [x2] [y2]
ASCII 254 "l" [x1] [y1] [x2] [y2]

Parameters	Parameter	Length	Description
	x1	1	Left bounds.
	y1	1	Top Bounds.
	x2	1	Right Bounds.
	y2	1	Bottom Bounds.

Description This command will draw a line from (x1,y1) to (x2,y2) using the current drawing color. Lines may be drawn from any part of the display to any other part. However, it may be important to note that the line may interpolate differently right to left, or left to right. This means that a line drawn in white from right to left may not fully erase the same line drawn in black from left to right.

Remembered No

8.5 Continue a Line

Syntax	Hexadecimal	0xFE 0x65 [x] [y]	
	Decimal	254 101 [x] [y]	
	ASCII	254 "e" [x] [y]	
Parameters	Parameter	Length	Description
	x	1	Left bounds.
	y	1	Top Bounds.
Description	This command will draw a line with the current drawing color from the last line end (x2,y2) to (x,y). This command uses the global drawing color.		
Remembered	No		

8.6 Draw a Rectangle

Syntax	Hexadecimal	0xFE 0x72 [color] [x1] [y1] [x2] [y2]	
	Decimal	254 114 [color] [x1] [y1] [x2] [y2]	
	ASCII	254 "r" [color] [x1] [y1] [x2] [y2]	
Parameters	Parameter	Length	Description
	color	1	Drawing color (0: White, 1-255: Black).
	x1	1	Left bounds.
	y1	1	Top Bounds.
	x2	1	Right Bounds.
	y2	1	Bottom Bounds.
Description	This command draws a rectangular box in the specified color (0: White, 1: Black). The top left corner is specified by (x1,y1) and the bottom right corner by (x2,y2).		
Remembered	No		

8.7 Drawing a Solid Rectangle

Syntax	Hexadecimal	0xFE 0x78 [color] [x1] [y1] [x2] [y2]	
	Decimal	254 120 [color] [x1] [y1] [x2] [y2]	
	ASCII	254 "x" [color] [x1] [y1] [x2] [y2]	
Parameters	Parameter	Length	Description
	color	1	Drawing color (0: White, 1-255: Black).
	x1	1	Left bounds.
	y1	1	Top Bounds.
	x2	1	Right Bounds.
	y2	1	Bottom Bounds.

Description	This command draws a solid rectangle in the specified color (0: White, 1: Black). The top left corner is specified by (x1,y1) and the bottom right corner by (x2,y2). Since this command involves considerable processing overhead, we strongly recommend the use of flow control, particularly if the command is to be repeated frequently.
Remembered	No

8.8 Initializing a Bar Graph

Syntax	Hexadecimal 0xFE 0x67 [refID] [type] [x1] [y1] [x2] [y2] Decimal 254 103 [refID] [type] [x1] [y1] [x2] [y2] ASCII 254 "g" [refID] [type] [x1] [y1] [x2] [y2]																					
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Length</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>refID</td> <td>1</td> <td>Reference number</td> </tr> <tr> <td>type</td> <td>1</td> <td>Type of bar graph.</td> </tr> <tr> <td>x1</td> <td>1</td> <td>Left bounds.</td> </tr> <tr> <td>y1</td> <td>1</td> <td>Top Bounds.</td> </tr> <tr> <td>x2</td> <td>1</td> <td>Right Bounds.</td> </tr> <tr> <td>y2</td> <td>1</td> <td>Bottom Bounds.</td> </tr> </tbody> </table>	Parameter	Length	Description	refID	1	Reference number	type	1	Type of bar graph.	x1	1	Left bounds.	y1	1	Top Bounds.	x2	1	Right Bounds.	y2	1	Bottom Bounds.
Parameter	Length	Description																				
refID	1	Reference number																				
type	1	Type of bar graph.																				
x1	1	Left bounds.																				
y1	1	Top Bounds.																				
x2	1	Right Bounds.																				
y2	1	Bottom Bounds.																				
Description	<p>This command initializes a bar graph referred to by number [reference number] of type [type] with size from (x1,y1) (top left) to (x2,y2) (bottom right). A maximum of 16 bar graphs with reference numbers from 0 to 15 can be initialized as:</p> <table border="1"> <thead> <tr> <th>[type]</th> <th>Direction</th> <th>Bar Start Point</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Vertical</td> <td>Bottom</td> </tr> <tr> <td>1</td> <td>Horizontal</td> <td>Left</td> </tr> <tr> <td>2</td> <td>Vertical</td> <td>Top</td> </tr> <tr> <td>3</td> <td>Horizontal</td> <td>Right</td> </tr> </tbody> </table> <p>The bar graphs may be located anywhere on the display, but if they overlap, they will not display properly. It is important that [x1] is less than [x2], and [y1] is less than [y2]. This command doesn't actually draw the graph, it must be filled in using the Fill Bar Graph command. The unit saves time by only drawing that part of the bar graph which has changed from the last write, so the representation on the screen may not survive a screen clear or other corrupting action. A write of value zero, followed by new values will restore the proper look of the bar graph.</p>	[type]	Direction	Bar Start Point	0	Vertical	Bottom	1	Horizontal	Left	2	Vertical	Top	3	Horizontal	Right						
[type]	Direction	Bar Start Point																				
0	Vertical	Bottom																				
1	Horizontal	Left																				
2	Vertical	Top																				
3	Horizontal	Right																				
Remembered	No																					

8.9 Drawing a Bar Graph

Syntax	Hexadecimal	0xFE 0x69 [ref] [value]	
	Decimal	254 105 [ref] [value]	
	ASCII	254 "i" [ref] [value]	
Parameters	Parameter	Length	Description
	ref	1	Initialized bar graph reference number.
Description	value	1	The number of pixels to fill.
	Once the bar graph has been initialized it can be filled in using this command. This command sets the bar graph specified by the [ref] number to fill in [value]. [value] is given in pixels and should not exceed the available height/width of the graph. (If it does the graph will simply be written to its maximum size.)		
Remembered	No		

8.10 Initializing a Strip Chart

Syntax	Hexadecimal	0xFE 0x6A [refID] [x1] [y1] [x2] [y2]	
	Decimal	254 106 [refID] [x1] [y1] [x2] [y2]	
	ASCII	254 "j" [refID] [x1] [y1] [x2] [y2]	
Parameters	Parameter	Length	Description
	refID	1	Reference number
	x1	1	Left bounds.
	y1	1	Top Bounds.
	x2	1	Right Bounds.
	y2	1	Bottom Bounds.

Description A strip chart is an area of the screen reserved for horizontal scrolling. This is normally used as follows:

- Initialize the strip chart, which reserves the appropriate area of the screen.
- Draw a line segment at the right or left side of the strip chart.
- Shift the strip chart to the right or left.
- Draw the next line segment.
- Used this way the strip chart can produce a graph which scrolls smoothly horizontally in either direction. With text the strip chart can produce a marquis effect.

NOTE If the strip chart is used with text we recommend the use of a 6 or 7 pixel wide fixed width character set, with each character placed 8 pixels from the start of the previous one.

Up to 7 strip charts ([ref] = 0 - 6) may be defined. To initialize a strip chart the user must define an area on the display in which to place the strip chart. (x1,y1) is the top left corner of the area to be used, where [x1] is the placement of the column where the strip chart is to begin and [y1] is the row. The user must then define [x2] as the bottom right column of the area to be utilized and [y2] as the bottom right row.

NOTE The definition of x must lie on byte boundaries. That is, x must be defined as 0x00, 0x08, 0x10, etc. This restriction does not apply to y values.

Remembered No

8.11 Shifting a Strip Chart

Syntax	Hexadecimal 0xFE 0x6B [ref] Decimal 254 107 [ref] ASCII 254 "k" [ref]						
Parameters	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="text-align: left;">Parameter</th> <th style="text-align: left;">Length</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>ref</td> <td style="text-align: center;">1</td> <td>Reference number of a strip chart that has already been created.</td> </tr> </tbody> </table>	Parameter	Length	Description	ref	1	Reference number of a strip chart that has already been created.
Parameter	Length	Description					
ref	1	Reference number of a strip chart that has already been created.					

Description	<p>This command shifts the strip chart left or right. [ref] determines both which strip chart is used and which direction it will shift. The direction is selected by the most significant bit (MSB):</p> <ul style="list-style-type: none"> • MSB: 0 shifts left • MSB: 1 shifts right <p>For example if [ref] is 1:</p> <ul style="list-style-type: none"> • 254 107 1 (hex FE 6B 01) shifts left • 254 107 129 (hex FE 6B 81) shifts right <p>This command shifts the contents of the area defined in the Initialize Strip Chart command 8 pixels at a time.</p>
Remembered	No

9 General Purpose Output

9.1 Introduction

General purpose outputs allow you to connect devices, such as LEDs, to the and supply them with up to 20mA of current at 5V. The has 1 GPO which are software controlled, with functions to turn it on/off and set the power state for the next startup.

9.2 General Purpose Output Off

Syntax	Hexadecimal	0xFE 0x56 [Num]	
	Decimal	254 86 [Num]	
	ASCII	254 "V" [Num]	
Parameters	Parameter	Length	Description
	Num	1	GPO number.
Description	This command turns OFF general purpose output [num].		

NOTE OFF means that the output is pulled HIGH.

Remembered	Yes
------------	-----

9.3 General Purpose Output On

Syntax	Hexadecimal	0xFE 0x57 [Num]	
	Decimal	254 87 [Num]	
	ASCII	254 “W” [Num]	
Parameters	Parameter	Length	Description
	Num	1	GPO number.
Description	This command turns ON general purpose output [num]. The standard GPO’s on the output 20mA of current at 5V.		

NOTE ON means the output is pulled LOW.

Remembered Yes

9.4 Set Startup GPO state

Syntax	Hexadecimal	0xFE 0xC3 [Num] [state]	
	Decimal	254 195 [Num] [state]	
Parameters	Parameter	Length	Description
	Num	1	GPO number.
	state	1	Startup state (0: Off, 1: On)
Description	This command will set the startup state for the GPO on the next power up. A value of one will cause the GPO to be off on the next startup while a value of one will cause the GPO to be on.		

NOTE This command does not affect the current state of the GPO.

Remembered Always

10 Display Functions

10.1 Introduction

The employs software controlled display settings, which allow for control over, clearing the screen, changing the brightness and contrast or setting timers for turning it on or off. The combination of these allow you complete software control over your display’s appearance.

10.2 Clear Screen

Syntax	Hexadecimal	0xFE 0x58
	Decimal	254 88
	ASCII	254 “X”
Description	This command clears the display and resets the text insertion position to the top left position of the screen defined in the font metrics.	
Remembered	No	

10.3 Display On

Syntax	Hexadecimal	0xFE 0x42 [min]	
	Decimal	254 66 [min]	
	ASCII	254 “B” [min]	
Parameters	Parameter	Length	Description
	min	1	Minutes before turning the display on (0 to 90).
Description	This command turns the backlight on after the [minutes] timer has expired, with a ninety minute maximum timer. A time of 0 specifies that the backlight should turn on immediately and stay on. When this command is sent while the remember function is on, the timer will reset and begin after power up.		
Remembered	Yes		
Default	0		

10.4 Display Off

Syntax	Hexadecimal	0xFE 0x46
	Decimal	254 70
	ASCII	254 “F”
Description	This command turns the backlight off immediately. The backlight will remain off until a ‘Display On’ command has been received.	
Remembered	Yes	

10.5 Set Brightness

Syntax	Hexadecimal	0xFE 0x99 [brightness]	
	Decimal	254 153 [brightness]	
Parameters	Parameter	Length	Description
	brightness	1	Display brightness setting (0 to 255).

Description	This command sets the display [brightness]. If the remember function is on, this command acts the same as 'Set and Save Brightness'.
Remembered	Yes
Default	255

10.6 Set and Save Brightness

Syntax	Hexadecimal	0xFE 0x98 [brightness]	
	Decimal	254 152 [brightness]	
Parameters	Parameter	Length	Description
	brightness	1	Backlight setting (0 to 255).
Description	This command sets and saves the display [brightness] as default.		
Remembered	Always		

10.7 Set Contrast

Syntax	Hexadecimal	0xFE 0x50 [contrast]	
	Decimal	254 80 [contrast]	
	ASCII	254 "P" [contrast]	
Parameters	Parameter	Length	Description
	contrast	1	Contrast value (0 to 255).
Description	<p>This command sets the display's contrast to [contrast], where [contrast] is a value between 0x00 and 0xFF (between 0 to 255). Lower values cause 'on' elements in the display area to appear lighter, while higher values cause 'on' elements to appear darker. Lighting and temperature conditions will affect the actual value used for optimal viewing. Individual display modules will also differ slightly from each other in appearance. In addition, values for optimal viewing while the display backlight is on may differ from values used when backlight is off. This command does not save the [contrast] value, and is lost after power down; but this command has the option of remembering the settings when issued with the Remember function 'on'. When this is the case, this command is the same as the Set and Save Contrast command.</p>		
<hr/> <p>NOTE This command has only 32 levels for X-Board based displays, meaning eight contrast settings will have the same single effect. Effectively, values 0 through 7, 8 through 15, and so on will result in the same setting.</p> <hr/>			
Remembered	Yes		
Default	128		

10.8 Set and Save Contrast

Syntax	Hexadecimal	0xFE 0x91 [contrast]	
	Decimal	254 145 [contrast]	
Parameters	Parameter	Length	Description
	contrast	1	Contrast value (0 to 255).
Description	This command sets the display's contrast to [contrast], where [contrast] is a value between 0x00 and 0xFF (between 0 to 255). Lower values cause 'on' elements in the display area to appear lighter, while higher values cause 'on' elements to appear darker. Lighting conditions will affect the actual value used for optimal viewing. Individual display modules will also differ slightly from each other in appearance. In addition, values for optimal viewing while the display backlight is on may differ from values used when backlight is off.		

NOTE This command saves the [contrast] value so that it is not lost after power down.

Remembered	Yes
Default	128

11 Filesystem

11.1 Introduction

The incorporates a 16 Kbyte on board flash memory in order to allow up to 128 font and bitmap files to be transferred directly onto the display and recalled whenever necessary. The filesystem can address font and bitmap files combined up to 16 Kbytes. It is recommended that fonts and bitmaps are uploaded when possible all together after a filesystem wipe to preserve memory integrity. These fonts and bitmaps can then be locked to ensure they remain intact. This section covers uploading, downloading, deleting and moving files, as well as getting the remaining space or wiping the filesystem.

11.1.1 File Upload Protocol

In order to allow fonts and bitmaps to be uploaded to the on board flash memory Matrix Orbital has developed a simple protocol that supports RS-232/TTL or I²C communications. In order to begin a file transmission the first step will be to provide the display module with the appropriate command bytes, meaning the command prefix, 0xFE, followed by the command number, 0x24 for a font file, or 0x5E for a bitmap file. This will begin the file transfer sequence. The next step will be to request a reference identification number (ref ID) which will allow you to identify the file for future use. Reference ID numbers can be any byte between 0x00 and 0x7F, however each ID must be unique.

The next part of uploading a font file is to provide the display module with the two byte file size of the

Host	Display	Comments
0xFE		Command Prefix
0x24		Upload Font File Command
0x01		Reference ID
0x19		Size (LSB)
0x00		Size (MSB)
	0x01	Confirmation Byte
0x01		Confirmation Byte
0x05		Font Width
	0x05	Echo Font Width
0x01		Confirmation Byte
0x07		Font Height
	0x07	Echo Font Height
0x01		Confirmation Byte
0x49		Font ASCII Start Value
	0x49	Echo Font ASCII Start Value
0x01		Confirmation Byte
...
0x60		Last Font File Byte
	0x60	Echo Last Font File Byte
0x01		Confirm Upload Finished

Table 43: Upload Protocol

data that you wish to transfer, LSB to MSB. The LSB must be transmitted first followed by the MSB. After receiving the MSB the display module will send a confirm byte, 0x01, if the file fits and continue, or decline byte, 0x08, and terminate the session.

Byte	Description
0x01	Confirm: Will continue the file transfer.
0x08	Decline: Terminate the session.

The last part of uploading a font file is to upload the file data. After transmitting each byte of the file the module will echo the byte and wait for a confirmation byte of 0x01 until the file has completed uploading. Below is an example of uploading the font file which we created in *Section 5.1.2 on page 18*.

At times that the display or the host sees anything else other than 0x01 for confirmation (usually a 0x08) the upload is aborted.

NOTES

- The has watch dog timer, set to 2.1 seconds in between transmissions, in order prevent the display module from staying in a waiting state.
 - Once the timeout has been reached the timer will reset the display and issue a 0xFE 0xD4 response to the host to signal that this has happened.
-

11.1.2 XModem Upload Protocol

In addition to its original simple upload format, Matrix Orbital has added an XModem based protocol. This facilitates much faster download speeds by increasing the packet size from 1 byte to 128 bytes greatly increasing throughput. A two byte CRC check is preformed at the end of each packet in place of the byte echo system seen in the original protocol. However, the overall protocol remains much the same as the original, but much faster.

To begin the upload, a series of command bytes are sent, much like the original protocol. However, no distinction is made between bitmap and font as the XModem protocol is used to upload bin or ebin files that contain all the bitmaps and fonts required for the unit. Once the command bytes are sent, the size of the file is sent in two bytes, least significant byte first. Then two additional bytes are sent of the value zero.

At this point the display will respond with an ACK if the file fits, or a NAK otherwise. Please note that these values are different than those of the original protocol as seen in the table below. If a NAK is seen at any point by the host, the upload is to be aborted in the same fashion as the regular protocol.

If the file will fit, the start of header byte will be sent by the host, followed by a block count representing the number of 128 byte blocks remaining to upload in regular and inverted forms. The display will then check to make sure the block count value matches its own before ACKing. The host can then send a 128 byte block of data followed by that blocks high and low CRC16 bytes. The display then preforms a CRC check on the data receive and ACKs if it matches that which was sent. Transfer continues with a block count and continues in this way until the end of file is reached.

Once the end of the upload file is reached, the host should transmit a single end of transmission byte. If the end of file is expected, the display will ACK one last time. This EOT byte along with the other special characters mentioned above is listed in the table below.

Character	Byte	Description
ACK	0x06	Acknowledged; successful data transmission
NAK	0x21	Not Acknowledged; transmission unsuccessful, abort upload
SOH	0x01	Start of Header; begin upload process
EOT	0x04	End of Transmission; file upload complete

Below is an example of uploading a bin or ebin file using the XModem protocol.

11.2 Wipe Filesystem

Host	Display	Comments
0xFE		Command Prefix
0xDB		XModem Upload Command
0x85		Command byte 1
0x06		Command byte 2
0x30		Command byte 3
0x00		Size Low Byte
0x40		Size High Byte
0x00		0
0x00		0
	0x06	ACK (NAK if file is too big)
0x01		Start of Header
0x80		Block Count
0x7F		255 - Block Count
	0x06	ACK (NAK if counts don't match)
<128 bytes>		Data Block
0x1E		CRC High Byte
0x47		CRC Low Byte
	0x06	ACK (NAK if CRCs don't match)
0x7F		Block Count
0x80		255 - Block Count
	0x06	ACK (NAK if counts don't match)
<128 bytes>		Data Block
0x5A		CRC High Byte
0x0D		CRC Low Byte
	0x06	ACK (NAK if CRCs don't match)
...
0x04		End of Transmission
	0x06	ACK (NAK if EOT is not expected)

Table 44: XModem Upload Protocol

Syntax	Hexadecimal	0xFE 0x21 0x59 0x21
	Decimal	254 33 89 33
	ASCII	254 “!” “Y” “!”
Description	This command completely erases the display’s non-volatile memory. It removes all fonts, font metrics, bitmaps, and settings (current font, cursor position, communication speed, etc.). It is an “odd” command in that it is three bytes in length in order to prevent accidental execution.	

NOTE After deleting the file system it is important to cycle power to your display to ensure the removal process is completed.

Remembered Yes

11.3 Deleting a File

Syntax	Hexadecimal	0xFE 0xAD [type] [refID]	
	Decimal	254 173 [type] [refID]	
Parameters	Parameter	Length	Description
	type	1	Type of file (0:Font, 1:Bitmap)
	refID	1	Reference ID of the file to delete.
Description	This command erases a single file at a time within the memory when given two parameters: [type] and [refID]. The file type and reference number are defined when the file is saved to the .		

- [type] = 1: Bitmap
- [type] = 0: Font

NOTE After deleting a file it is important to cycle power to your display to ensure file system integrity.

Remembered Yes

11.4 Get Filesystem Space

Syntax	Hexadecimal	0xFE 0xAF
	Decimal	254 175
Description	This command will return 4 bytes, LSB to MSB for how many bytes are remaining in the 16 KB on board memory.	

Remembered No

11.5 Get Filesystem Directory

Syntax	Hexadecimal 0xFE 0xB3 Decimal 254 179
Description	This command will return a directory of the contents of the file system. The first byte returned will be a hex value representing the number of entries in the filesystem, followed by four bytes for each entry. See the following tables:

Filesystem Header	
Bytes	Description
1	Hex value representing the number of entries in the filesystem

File Entry	
Bytes	Description
1	Flag: Hex value of 0x00 indicates that this file entry has not been used.
1	FileID/Type: 1st bit is the file type (0: Font, 1: Bitmap). Next 7 bits are the file ID.
1	File Size: LSB
1	File Size: MSB

Remembered No

11.6 Filesystem Upload

Syntax	Hexadecimal 0xFE 0xB0 [Size] [Data] Decimal 254 176 [Size] [Data]									
Parameters	<table border="1"><thead><tr><th>Parameter</th><th>Length</th><th>Description</th></tr></thead><tbody><tr><td>Size</td><td>4</td><td>LSB to MSB filesystem image data</td></tr><tr><td>Data</td><td>var</td><td>Actual data to upload</td></tr></tbody></table>	Parameter	Length	Description	Size	4	LSB to MSB filesystem image data	Data	var	Actual data to upload
Parameter	Length	Description								
Size	4	LSB to MSB filesystem image data								
Data	var	Actual data to upload								
Description	This command will upload a filesystem image, LSB to MSB to the display (16KB). The size used is almost always the entire 16kB, meaning the values 0x00 0x40 0x00 0x00 must be issued. After which the filesystem data can be uploaded LSB to MSB in the same manner as a font or bitmap file.									

Remembered Always

11.7 Downloading a File

Syntax	Hexadecimal	0xFE 0xB2 [Type] [refID]	
	Decimal	254 178 [Type] [refID]	
Parameters	Parameter	Length	Description
	Type	1	File type (0:Font File, 1:Bitmap)
	refID	1	Reference ID number
Description	Download a specified file from the filesystem. The first 4 bytes will be the length of the file (LSB to MSB) followed by 2 bytes representing the width and height of the image then the data contained in the file.		
Remembered	No		

11.8 Moving a File

Syntax	Hexadecimal	0xFE 0xB4 [oldT] [oldID] [newT] [newID]		
	Decimal	254 180 [oldT] [oldID] [newT] [newID]		
Parameters	Parameter	Length	Description	
	oldT	1	Old file type	
	oldID	1	Old file ID	
	newT	1	New file type	
	newID	1	New file ID	
Description	This command can be used to move a file to a new file ID, or correct the type of a file that was uploaded incorrectly. The command first checks to see if there is a file identified by [oldT] and [oldID]. If it does exist, and there is no file already with the desired type and ID, the ID and type of the old file will be changed to [newT] and [newID] respectively.			
Remembered	Always			

12 Data Security

12.1 Introduction

Ensuring that your display's exactly what you want it to can be the difference between a projects success and failure. This is why we incorporate features such as Data Lock into the With this new feature you now are in control over of how and when settings will be changed so there is no need to worry about the module acting exactly like you expected it to because all the settings may be locked and remembered for the next power up.

12.2 Set Remember

Syntax	Hexadecimal	0xFE 0x93 [switch]	
	Decimal	254 147 [switch]	
Parameters	Parameter	Length	Description
	switch	1	0: Do not remember, 1: Remember
Description	This command allows you to switch the remember function on and off. To use the remember function, set remember to on, then set all of the settings that you wish to save, settings that are listed as 'Remember: Yes' support being saved into the non-volatile memory. After you have set all of the commands that you wish to save, you may then cycle the power and check the display settings to ensure that all the settings have been saved. If you wish to use remember again after cycling the power, you must set it to on again.		

NOTES

- Writing to non-volatile memory is time consuming and slows down the operation of the display.
- Non-volatile memory has a 'write limit' and may only be changed approximately 100,000 times.

Remembered	No
Default	Do not remember

12.3 Data Lock

Syntax	Hexadecimal	0xFE 0xCA 0xF5 0xA0 [level]	
	Decimal	254 202 245 160 [level]	
Parameters	Parameter	Length	Description
	level	1	Sets the data lock level

Description

Paranoia allows you to lock the module from displaying information, as well as enables the protection of the filesystem and module settings.

Each bit corresponds corresponds to a different lock level, while sending a zero will unlock your display as the following tables explains:

Bit	Data Lock Level	Description
0-2	Reserved	Should be left 0
3	Communication Speed Lock	When this bit is set (1) the Baud Rate and I ² C Slave address are locked
4	Setting Lock	When this bit is set (1) the display settings such as backlight, contrast and GPO settings are locked. (Internal EEPROM)
5	Filesystem Lock	When this bit is set (1) the external EEPROM is locked, this has the same effect as the File System Jumper
6	Command Lock	When this bit is set (1) all commands but commands 202/203 are locked. (cmd lock)
7	Display Lock	When this bit is set (1) the module is locked from displaying any new information. (text lock)

NOTES

- Sending a new data lock level will override the previous data lock level.
 - Data lock levels may be combined.
-

Remembered
Default
Examples

Always

0

Hex	Dec	Binary	Description
0x00	0	0	Unlock
0x50	80	01010000	Setting and Command Lock

12.4 Set and Save Data Lock

Syntax	Hexadecimal	0xFE 0xCB 0xF5 0xA0 [level]	
	Decimal	254 203 245 160 [level]	
Parameters	Parameter	Length	Description
	level	1	Sets the data lock level
Description	This command will set and save the data lock level. See the Data Lock section for more information.		
Remembered	Always		
Default	0		

12.5 Dump the Filesystem

Syntax	Hexadecimal	0xFE 0x30	
	Decimal	254 48	
	ASCII	254 "0"	
Description	This will allow you to dump the filesystem for debugging purposes. It will return a 4 byte value LSB to MSB followed by 16384 bytes making up the file system.		
Remembered	No		

12.6 Write Customer Data

Syntax	Hexadecimal	0xFE 0x34 [data]	
	Decimal	254 52 [data]	
	ASCII	254 "4" [data]	
Parameters	Parameter	Length	Description
	data	16	Writes the customer data
Description	Writes the customer Data. 16 Bytes of data can be saved in non-volatile memory.		
Remembered	No		

12.7 Read Customer Data

Syntax	Hexadecimal	0xFE 0x35
	Decimal	254 53
	ASCII	254 "5"

Description Reads whatever was written by Write Customer Data.
Remembered No

13 Miscellaneous

13.1 Introduction

This chapter covers the 'Report Version Number' and 'Read Module Type' commands. These commands can be particularly useful to find out more information about the display module before contacting technical support.

13.2 Read Version Number

Syntax Hexadecimal 0xFE 0x36
Decimal 254 54
ASCII 254 "6"
Description This command will return a byte representing the version of the module, see the following table as an example:

Hex Value	Version Number
0x19	Version 1.9
0x57	Version 5.7

Remembered No

13.3 Read Module Type

Syntax Hexadecimal 0xFE 0x37
Decimal 254 55
ASCII 254 "7"

Description

This command will return a hex value corresponding to the the model number of the module see the following table:

Hex	Product ID	Hex	Product ID
1	LCD0821	2	LCD2021
5	LCD2041	6	LCD4021
7	LCD4041	8	LK202-25
9	LK204-25	A	LK404-55
B	VFD2021	C	VFD2041
D	VFD4021	E	VK202-25
F	VK204-25	10	GLC12232
13	GLC24064	14	Unused
15	GLK24064-25	16	Unused
21	Unused	22	GLK12232-25
23	Unused	24	GLK12232-25-SM
25	GLK24064-16-1U-USB	26	GLK24064-16-1U
27	GLK19264-7T-1U-USB	28	GLK12232-16
29	GLK12232-16-SM	2A	GLK19264-7T-1U
2B	LK204-7T-1U	2C	LK204-7T-1U-USB
31	LK404-AT	32	MOS-AV-162A
33	LK402-12	34	LK162-12
35	LK204-25PC	36	LK202-24-USB
37	VK202-24-USB	38	LK204-24-USB
39	VK204-24-USB	3A	PK162-12
3B	VK162-12	3C	MOS-AP-162A
3D	PK202-25	3E	MOS-AL-162A
3F	MOS-AL-202A	40	MOS-AV-202A
41	MOS-AP-202A	42	PK202-24-USB
43	MOS-AL-082	44	MOS-AL-204
45	MOS-AV-204	46	MOS-AL-402
47	MOS-AV-402	48	LK082-12
49	VK402-12	4A	VK404-55
4B	LK402-25	4C	VK402-25
4D	PK204-25	4E	Unused
4F	MOS	50	MOI
51	XBoard-S	52	XBoard-I
53	MOU	54	XBoard-U
55	LK202-25-USB	56	VK202-25-USB
57	LK204-25-USB	58	VK204-25-USB
5B	LK162-12-TC	5C	Unused
71	Unused	72	GLK240128-25
73	LK404-25	74	VK404-25
77	Unused	78	GLT320240
79	GLT480282	7A	GLT240128

Remembered

No

14 Command Summary

14.1 Communications

Description	Syntax	Page
Turn Flow Control On	Hexadecimal	0xFE 0x3A [full] [empty]
	Decimal	254 58 [full] [empty]
	ASCII	254 “;” [full] [empty]
Turn Flow Control Off	Hexadecimal	0xFE 0x3B
	Decimal	254 59
	ASCII	254 “;”
Changing the I ² C Slave Address	Hexadecimal	0xFE 0x33 [adr]
	Decimal	254 51 [adr]
	ASCII	254 “3” [adr]
Changing the Baud Rate	Hexadecimal	0xFE 0x39 [speed]
	Decimal	254 57 [speed]
	ASCII	254 “9” [speed]
Setting a Non-Standard Baud Rate	Hexadecimal	0xFE 0xA4 [speed]
	Decimal	254 164 [speed]

14.2 Fonts

Description	Syntax	Page
Uploading a Font File	Hexadecimal	0xFE 0x24 [refID] [size] [data]
	Decimal	254 36 [refID] [size] [data]
	ASCII	254 “\$” [refID] [size] [data]
Setting the Current Font	Hexadecimal	0xFE 0x31 [refID]
	Decimal	254 49 [refID]
	ASCII	254 “1” [refID]
Font Metrics	Hexadecimal	0xFE 0x32 [lm] [tm] [csp] [lsp] [srow]
	Decimal	254 50 [lm] [tm] [csp] [lsp] [srow]
	ASCII	254 “2” [lm] [tm] [csp] [lsp] [srow]
Set Box Space Mode	Hexadecimal	0xFE 0xAC [value]
	Decimal	254 172 [value]

14.3 Text

Description	Syntax	Page
Move Cursor Home	Hexadecimal	0xFE 0x48
	Decimal	254 72
	ASCII	254 “H”

Description	Syntax	Page
Setting the Cursor Position	Hexadecimal	0xFE 0x47 [col] [row]
	Decimal	254 71 [col] [row]
	ASCII	254 “G” [col] [row]
Setting the Cursor Coordinate	Hexadecimal	0xFE 0x79 [x] [y]
	Decimal	254 121 [x] [y]
	ASCII	254 “y” [x] [y]
Auto Scroll On	Hexadecimal	0xFE 0x51
	Decimal	254 81
	ASCII	254 “Q”
Auto Scroll Off	Hexadecimal	0xFE 0x52
	Decimal	254 82
	ASCII	254 “R”

14.4 Bitmaps

Description	Syntax	Page
Uploading a Bitmap File	Hexadecimal	0xFE 0x5E [refID] [size] [data]
	Decimal	254 94 [refID] [size] [data]
	ASCII	254 “^” [refID] [size] [data]
Drawing a Bitmap from Memory	Hexadecimal	0xFE 0x62 [refID] [X] [Y]
	Decimal	254 98 [refID] [X] [Y]
	ASCII	254 “b” [refID] [X] [Y]
Drawing a Bitmap Directly	Hexadecimal	0xFE 0x64 [X] [Y] [W] [H] [D]
	Decimal	254 100 [X] [Y] [W] [H] [D]
	ASCII	254 “d” [X] [Y] [W] [H] [D]

14.5 Bar Graphs and Drawing

Description	Syntax	Page
Set Drawing Color	Hexadecimal	0xFE 0x63 [color]
	Decimal	254 99 [color]
	ASCII	254 “c” [color]
Draw Pixel	Hexadecimal	0xFE 0x70 [x] [y]
	Decimal	254 112 [x] [y]
	ASCII	254 “p” [x] [y]
Drawing a Line	Hexadecimal	0xFE 0x6C [x1] [y1] [x2] [y2]
	Decimal	254 108 [x1] [y1] [x2] [y2]
	ASCII	254 “l” [x1] [y1] [x2] [y2]
Continue a Line	Hexadecimal	0xFE 0x65 [x] [y]
	Decimal	254 101 [x] [y]
	ASCII	254 “e” [x] [y]

Description	Syntax	Page
Draw a Rectangle	Hexadecimal	0xFE 0x72 [color] [x1] [y1] [x2] [y2]
	Decimal	254 114 [color] [x1] [y1] [x2] [y2]
	ASCII	254 “r” [color] [x1] [y1] [x2] [y2]
Drawing a Solid Rectangle	Hexadecimal	0xFE 0x78 [color] [x1] [y1] [x2] [y2]
	Decimal	254 120 [color] [x1] [y1] [x2] [y2]
	ASCII	254 “x” [color] [x1] [y1] [x2] [y2]
Initializing a Bar Graph	Hexadecimal	0xFE 0x67 [refID] [type] [x1] [y1] [x2] [y2]
	Decimal	254 103 [refID] [type] [x1] [y1] [x2] [y2]
	ASCII	254 “g” [refID] [type] [x1] [y1] [x2] [y2]
Drawing a Bar Graph	Hexadecimal	0xFE 0x69 [ref] [value]
	Decimal	254 105 [ref] [value]
	ASCII	254 “i” [ref] [value]
Initializing a Strip Chart	Hexadecimal	0xFE 0x6A [refID] [x1] [y1] [x2] [y2]
	Decimal	254 106 [refID] [x1] [y1] [x2] [y2]
	ASCII	254 “j” [refID] [x1] [y1] [x2] [y2]
Shifting a Strip Chart	Hexadecimal	0xFE 0x6B [ref]
	Decimal	254 107 [ref]
	ASCII	254 “k” [ref]

14.6 General Purpose Output

Description	Syntax	Page
General Purpose Output Off	Hexadecimal	0xFE 0x56 [Num]
	Decimal	254 86 [Num]
	ASCII	254 “V” [Num]
General Purpose Output On	Hexadecimal	0xFE 0x57 [Num]
	Decimal	254 87 [Num]
	ASCII	254 “W” [Num]
Set Startup GPO state	Hexadecimal	0xFE 0xC3 [Num] [state]
	Decimal	254 195 [Num] [state]

14.7 Display Functions

Description	Syntax	Page
Clear Screen	Hexadecimal	0xFE 0x58
	Decimal	254 88
	ASCII	254 “X”
Display On	Hexadecimal	0xFE 0x42 [min]
	Decimal	254 66 [min]
	ASCII	254 “B” [min]
Display Off	Hexadecimal	0xFE 0x46
	Decimal	254 70
	ASCII	254 “F”

Description	Syntax	Page
Set Brightness	Hexadecimal	0xFE 0x99 [brightness]
	Decimal	254 153 [brightness]
Set and Save Brightness	Hexadecimal	0xFE 0x98 [brightness]
	Decimal	254 152 [brightness]
Set Contrast	Hexadecimal	0xFE 0x50 [contrast]
	Decimal	254 80 [contrast]
	ASCII	254 "P" [contrast]
Set and Save Contrast	Hexadecimal	0xFE 0x91 [contrast]
	Decimal	254 145 [contrast]

14.8 Filesystem

Description	Syntax	Page
Wipe Filesystem	Hexadecimal	0xFE 0x21 0x59 0x21
	Decimal	254 33 89 33
	ASCII	254 "!" "Y" "!"
Deleting a File	Hexadecimal	0xFE 0xAD [type] [refID]
	Decimal	254 173 [type] [refID]
Get Filesystem Space	Hexadecimal	0xFE 0xAF
	Decimal	254 175
Get Filesystem Directory	Hexadecimal	0xFE 0xB3
	Decimal	254 179
Filesystem Upload	Hexadecimal	0xFE 0xB0 [Size] [Data]
	Decimal	254 176 [Size] [Data]
Downloading a File	Hexadecimal	0xFE 0xB2 [Type] [refID]
	Decimal	254 178 [Type] [refID]
Moving a File	Hexadecimal	0xFE 0xB4 [oldT] [oldID] [newT] [newID]
	Decimal	254 180 [oldT] [oldID] [newT] [newID]

14.9 Data Security

Description	Syntax	Page
Set Remember	Hexadecimal	0xFE 0x93 [switch]
	Decimal	254 147 [switch]
Data Lock	Hexadecimal	0xFE 0xCA 0xF5 0xA0 [level]
	Decimal	254 202 245 160 [level]
Set and Save Data Lock	Hexadecimal	0xFE 0xCB 0xF5 0xA0 [level]
	Decimal	254 203 245 160 [level]
Dump the Filesystem	Hexadecimal	0xFE 0x30
	Decimal	254 48
	ASCII	254 "0"

Description	Syntax	Page
Write Customer Data	Hexadecimal 0xFE 0x34 [data]	45
	Decimal 254 52 [data]	
	ASCII 254 “4” [data]	
Read Customer Data	Hexadecimal 0xFE 0x35	45
	Decimal 254 53	
	ASCII 254 “5”	

14.10 Miscellaneous

Description	Syntax	Page
Read Version Number	Hexadecimal 0xFE 0x36	46
	Decimal 254 54	
	ASCII 254 “6”	
Read Module Type	Hexadecimal 0xFE 0x37	46
	Decimal 254 55	
	ASCII 254 “7”	

14.11 Command By Number

Command Hex	Description Dec	Page ASCII		
0x21	33	“!”	Wipe Filesystem	38
0x24	36	“\$”	Uploading a Font File	20
0x30	48	“0”	Dump the Filesystem	45
0x31	49	“1”	Setting the Current Font	20
0x32	50	“2”	Font Metrics	21
0x33	51	“3”	Changing the I ² C Slave Address	14
0x34	52	“4”	Write Customer Data	45
0x35	53	“5”	Read Customer Data	45
0x36	54	“6”	Read Version Number	46
0x37	55	“7”	Read Module Type	46
0x39	57	“9”	Changing the Baud Rate	15
0x3A	58	“:”	Turn Flow Control On	13
0x3B	59	“;”	Turn Flow Control Off	14
0x42	66	“B”	Display On	34
0x46	70	“F”	Display Off	34
0x47	71	“G”	Setting the Cursor Position	23
0x48	72	“H”	Move Cursor Home	22
0x50	80	“P”	Set Contrast	35
0x51	81	“Q”	Auto Scroll On	23
0x52	82	“R”	Auto Scroll Off	24
0x56	86	“V”	General Purpose Output Off	32

Command	Description	Page		
Hex	Dec	ASCII		
0x57	87	“W”	General Purpose Output On	32
0x58	88	“X”	Clear Screen	33
0x5E	94	“^”	Uploading a Bitmap File	24
0x62	98	“b”	Drawing a Bitmap from Memory	25
0x63	99	“c”	Set Drawing Color	26
0x64	100	“d”	Drawing a Bitmap Directly	25
0x65	101	“e”	Continue a Line	27
0x67	103	“g”	Initializing a Bar Graph	29
0x69	105	“i”	Drawing a Bar Graph	29
0x6A	106	“j”	Initializing a Strip Chart	30
0x6B	107	“k”	Shifting a Strip Chart	31
0x6C	108	“l”	Drawing a Line	27
0x70	112	“p”	Draw Pixel	27
0x72	114	“r”	Draw a Rectangle	28
0x78	120	“x”	Drawing a Solid Rectangle	28
0x79	121	“y”	Setting the Cursor Coordinate	23
0x91	145		Set and Save Contrast	36
0x93	147		Set Remember	43
0x98	152		Set and Save Brightness	35
0x99	153		Set Brightness	34
0xA4	164		Setting a Non-Standard Baud Rate	16
0xAC	172		Set Box Space Mode	21
0xAD	173		Deleting a File	40
0xAF	175		Get Filesystem Space	40
0xB0	176		Filesystem Upload	41
0xB2	178		Downloading a File	41
0xB3	179		Get Filesystem Directory	41
0xB4	180		Moving a File	42
0xC3	195		Set Startup GPO state	33

15 Appendix

15.1 Specifications

15.1.1 Environmental

Table 71: Environmental Specifications

	Standard Temperature	Extended Temperature
Operating Temperature	0°C to +50°C	-20°C to +70°C
Storage Temperature	-20°C to +70°C	-30°C to +80°C
Operating Relative Humidity	90% max non-condensing	
Vibration (Operating)	4.9 m/s ² XYZ directions	
Vibration (Non-Operating)	19.6 m/s ² XYZ directions	
Shock (Operating)	29.4 m/s ² XYZ directions	
Shock (Non-Operating)	490 m/s ² XYZ directions	

15.1.2 Electrical

Table 72: Electrical Specifications

	Standard	Wide Voltage (V)	Wide Voltage with Efficient Switching Power Supply (VPT)
Supply Voltage	+5Vdc ±0.25V	+9V to +15V	+9V to +35V
Minimum Current	45mA typical		
Backlight On (YG)	add 220mA (265mA) typical		
Backlight On (GW & WB)	add 50mA (95mA) typical		
GPO	add up to 20mA		

15.2 Optical Characteristics

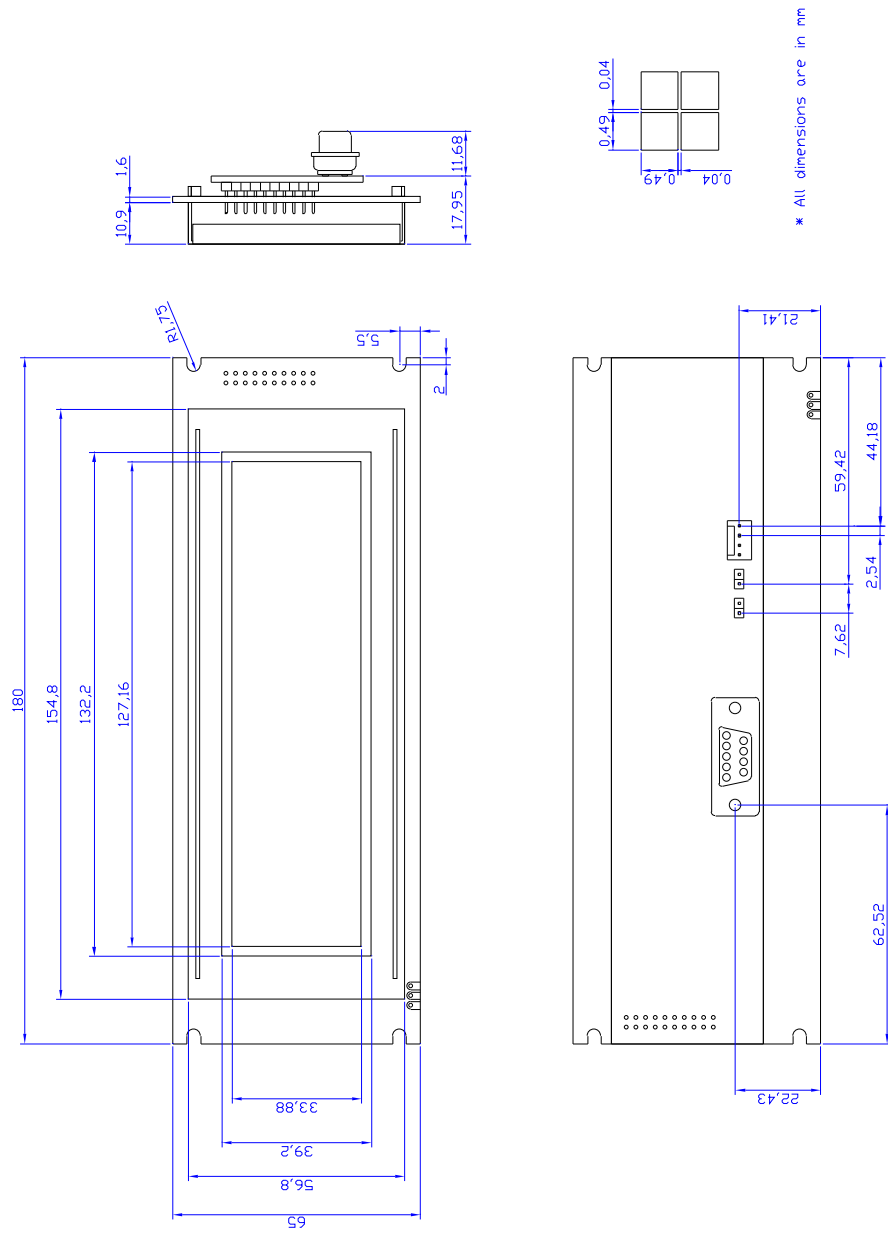
Table 73: Optical Characteristics

Pixel Layout	240 x 64 pixels XxY
Number of Characters	320 (maximum 40 characters x 8 Lines with 5x7 font)
Display Area	127.16 x 33.88mm XxY
Dot Size	0.49 x 0.49mm (XxY)
Dot Pitch	0.53 x 0.53mm (XxY)
LED Backlight Half-Life (YG)	50,000 hours typical
LED Backlight Half-Life (GW & WB)	10,000 hours typical
Backlight	Yellow/Green LED or White LED

NOTE To prolong life, it is recommended that the backlight be turned off when the display is not in use.

15.3 Physical Layout

Figure 21: Physical Diagram



G	LC	240	64	-GW	-V	-E
1	2	3	4	5	6	7

Table 74: Part Numbering Scheme

#	Description	Options
1	Screen Type	G: Graphic
2	Display Technology	LC: Liquid Crystal Display
3	Width	240: Pixel Width Count
4	Height	64: Pixel Height Count
5	Colour (Text/Background)	NP: Standard Grey/Yellow-Green -GW: Grey/White -WB: White/Blue
6	Input Voltage	NP: Standard (4.75-5.25V) -V: Extended Voltage (9.00-15.0V) -VPT: Extended Voltage with ESPS (9.00-35.0V)
7	Temperature	NP: Standard (0°C to +50°C) -E: Extended Temperature (-20°C to +70°C)

Table 75: Part Options

15.4 Ordering Information

15.5 Definitions

E Extended Temperature (-20C to 70C)

VPT Wide Voltage with Efficient Switching Power Supply (+9 to +35Vdc)

V Wide Voltage (+9 to +15Vdc)

GW Grey Text / White Background

WB White Text / Blue Background

MSB Most Significant Byte

LSB Least Significant Byte

15.6 Contacting Matrix Orbital

Telephone

Sales: 1(403)229-2737

Support: 1(403)204-3750

Table 76: Revision History

Revision	Description	Author
3.0	Initial Manual	Matrix Orbital
3.1	Backlight Life Update	Clark

On The Web

Sales: <http://www.MatrixOrbital.com>

Support: <http://www.MatrixOrbital.ca>

Forums: <http://www.lcdforums.com>

15.7 Revision History