



**MOS-AL-162A**  
**Technical Manual**

**Revision: 1.0**

# Contents

<b>Contents</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What to Expect From the MOS-AL-162A . . . . .	1
1.2 What Not to Expect From the MOS-AL-162A . . . . .	1
1.3 Setup for Testing With a PC . . . . .	1
1.4 Setup for Testing With a Basic Stamp . . . . .	2
1.5 Trying Out the MOS-AL-162A . . . . .	3
1.6 Technical Support . . . . .	3
<b>2 Connections</b>	<b>4</b>
2.1 Connector Pinout . . . . .	4
2.1.1 Power Connections . . . . .	5
2.1.2 Five Volt Modules . . . . .	5
2.2 Communication Connection . . . . .	6
2.2.1 RS-232 Communications . . . . .	7
2.2.2 TTL Communications . . . . .	8
2.3 Set Baud Rates . . . . .	9
2.4 General Purpose Outputs . . . . .	9
<b>3 Issuing Commands</b>	<b>11</b>
3.1 Issuing Commands From a Basic Stamp . . . . .	12
3.2 On Numbers . . . . .	13
3.3 ASCII Characters . . . . .	14
3.4 Example Command . . . . .	15
<b>4 Text Commands</b>	<b>16</b>
4.1 Introduction . . . . .	16
4.2 The Built In Character Font . . . . .	16
4.3 Command List . . . . .	17
4.3.1 Auto Line Wrap On . . . . .	17
4.3.2 Auto Line Wrap Off . . . . .	18
4.3.3 Set Cursor Position . . . . .	18
4.3.4 Send Cursor Home . . . . .	18
4.3.5 Turn On Underline Cursor . . . . .	19
4.3.6 Turn Off Underline Cursor . . . . .	19
4.3.7 Cursor Left . . . . .	19
4.3.8 Cursor Right . . . . .	20
4.3.9 Auto Scroll On . . . . .	20
4.3.10 Auto Scroll Off . . . . .	20
4.3.11 Turn On Blinking Block Cursor . . . . .	20
4.3.12 Turn Off Blinking Block Cursor . . . . .	21
4.3.13 Clear Display . . . . .	21

<b>5</b>	<b>Bar Graphs and Special Characters</b>	<b>21</b>
5.1	Introduction . . . . .	21
5.2	Command List . . . . .	22
5.2.1	Draw Vertical Bar Graph . . . . .	22
5.2.2	Load Startup Screen . . . . .	22
5.2.3	Define Custom Character . . . . .	23
5.2.4	Initialize Horizontal Bar Graph . . . . .	25
5.2.5	Initialize Medium Digits . . . . .	25
5.2.6	Draw Medium Digits . . . . .	26
5.2.7	Initialize Narrow Vertical Bar Graph . . . . .	26
5.2.8	Initialize Wide Vertical Bar Graph . . . . .	27
5.2.9	Draw Horizontal Bar Graph . . . . .	27
5.2.10	Remember Custom Character . . . . .	28
<b>6</b>	<b>Display Functions</b>	<b>28</b>
6.1	Introduction . . . . .	28
6.2	Command List . . . . .	28
6.2.1	Backlight On . . . . .	28
6.2.2	Backlight Off . . . . .	29
6.2.3	Set Contrast . . . . .	29
6.2.4	Set and Save Contrast . . . . .	30
6.2.5	Set and Save Backlight Brightness . . . . .	30
6.2.6	Set Backlight Brightness . . . . .	31
<b>7</b>	<b>GPO Functions</b>	<b>31</b>
7.1	Introduction . . . . .	31
7.2	Command List . . . . .	31
7.2.1	General Purpose Output Off . . . . .	31
7.2.2	General Purpose Output On . . . . .	31
7.2.3	Remember GPO . . . . .	32
<b>8</b>	<b>Communications Section</b>	<b>32</b>
8.1	Introduction . . . . .	32
8.2	Command List . . . . .	33
8.2.1	Enter Flow Control Mode . . . . .	33
8.2.2	Exit Flow Control Mode . . . . .	33
<b>9</b>	<b>Miscellaneous Commands</b>	<b>34</b>
9.1	Introduction . . . . .	34
9.2	Command List . . . . .	34
9.2.1	Set Serial Number . . . . .	34
9.2.2	Read Serial Number . . . . .	34
9.2.3	Read Version Number . . . . .	35
9.2.4	Read Module Type . . . . .	35
9.2.5	Remember . . . . .	35

<b>10 Command Summary</b>	<b>37</b>
10.1 Text Commands . . . . .	37
10.2 Bar Graphs and Special Characters . . . . .	38
10.3 Display Functions . . . . .	39
10.4 GPO Functions . . . . .	39
10.5 Communications Section . . . . .	40
10.6 Miscellaneous Commands . . . . .	40
<b>11 Appendix: Specifications and Options</b>	<b>41</b>
11.1 Specifications . . . . .	41
11.2 Options . . . . .	43
<b>12 Appendix: Glossary</b>	<b>44</b>

# 1 Introduction

The display is designed as the display unit for an associated controller. The controller may be anything from a single board, special purpose micro-controller to a PC, depending on the application. This controller is responsible for what is displayed on the screen of the display.

The display provides a simple command structure to allow text and bar graphs to be displayed on the screen. Text fonts are built in, and use standard ASCII mapping. Provision is made for up to 8 user defined characters.

The screen is luminous for low light situations. Display may be turned on or off under program control. Brightness is adjustable to compensate for differing lighting conditions.

General purpose outputs allow the controller to switch up to three electronic or electro-mechanical devices by issuing commands to the display unit. These can be used for controlling LEDs, relays, etc..

## 1.1 What to Expect From the MOS-AL-162A

The MOS-AL-162A comes equipped with the following features:

- 16 column by 2 line text LCD display
- Built in font with provision for up to 8 user defined characters
- 9600bps or a lighting fast 19.2 Kbps serial communication speed
- Communication over RS-232 or TTL
- Fully buffered so that no delays in transmission are necessary
- Ability to add a customized splash / startup screen
- Software controlled brightness with configurable time-out setting up to 90 minutes
- 32 levels of software controlled contrast
- Three general purpose outputs for a variety of applications
- Horizontal or vertical bar graphs
- Medium digit capability

## 1.2 What Not to Expect From the MOS-AL-162A

The display does not include bitmap graphics capability, except that permitted by defining special characters.

## 1.3 Setup for Testing With a PC

---

**NOTE** To test the module with a PC, the appropriate cable will need to be ordered or created accordingly.

---

Before setting up the application, the user may want to try out the display. When connected to a PC, the following will be required:

- A 4-pin power connector of the type used to connect 3.5" floppy drive. Take care not to connect the display to an unmodified spare power connector in a PC.

- A 5V power supply.
- A PC with a spare RS-232 port (COM1 or COM2).
- A custom cable is required for connection from the PC COM port to the display. This custom cable can be ordered directly from Matrix Orbital or from one of our distributors. The display connection is a four pin header. This header consists of power, Rx, Tx and ground as shown in Figure 4.

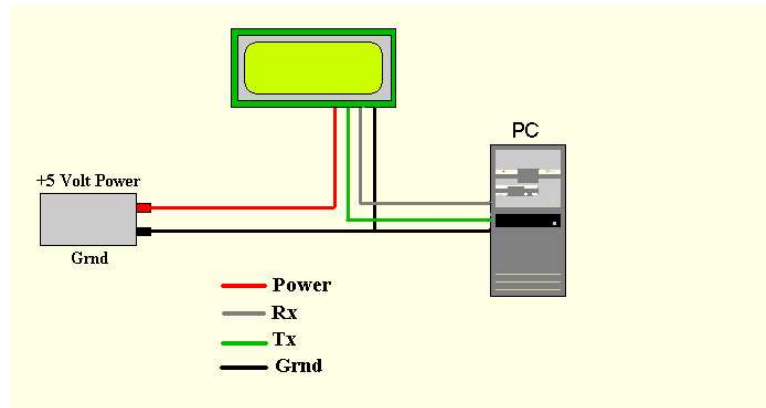


Figure 1: Connections for Testing

1. Refer to Figure 1 for the following steps.
2. Wire the connector to the power supply. On most connectors the RED lead will go to +5V and the BLACK lead to GND.



**WARNING** The Manufacturer's Warranty becomes void if the unit is subjected to over-voltage or reversed polarity.

3. Connect the display to the PC using the serial cable and adapter if required. Make sure the RS-232 cable includes the required ground lead. There must be no voltage differential between the RS-232 ground and the power supply ground.
4. Connect the power connector, making sure that the +5V goes to V+. Turn on the power: the module should come on and greet you with our company name and model number of the display.

## 1.4 Setup for Testing With a Basic Stamp

When connecting the module to the Basic Stamp development board, you will need to purchase the appropriate cable or modify a cable accordingly. Connection must be made via pins #2, #3 and #5 of the serial cable.

1. Connect pin #2 to P0
2. Connect pin #3 to P1

3. Connect pin #5 to Vss, which is ground

To connect the module with a development board, please see Figure 2. For code examples with the Basic Stamp, please see Section 3.1.

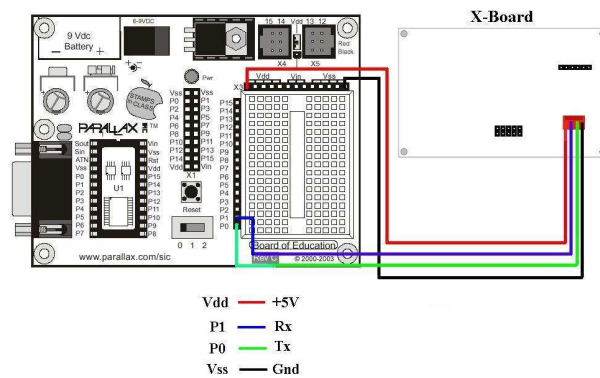


Figure 2: Interfacing to a Basic Stamp Development Board

## 1.5 Trying Out the MOS-AL-162A

The unit should be connected to power as in Section 1.3 . The PC and display should be on. To experiment with typing text, run a PC terminal program, such as Hyperterm. Make sure it's configured to use the correct com port. Set the baud rate to 19,200 as described in Section 2.3. If characters are typed on the keyboard, they should now appear on the display screen. Text will wrap around to the next line when the end of a line has been reached.

To exercise some of the other features of the display, a program (in any convenient language such as Basic or C) will need to be written in order to issue the required command strings. Most terminal programs are unable to issue the 0xFE character needed as a command prefix.

## 1.6 Technical Support

For technical support regarding this module, please see the following links provided:

<http://www.matrixorbital.com>

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

<http://www.parallax.com/>

## 2 Connections

### 2.1 Connector Pinout

Refer to Figure 3 for this section.

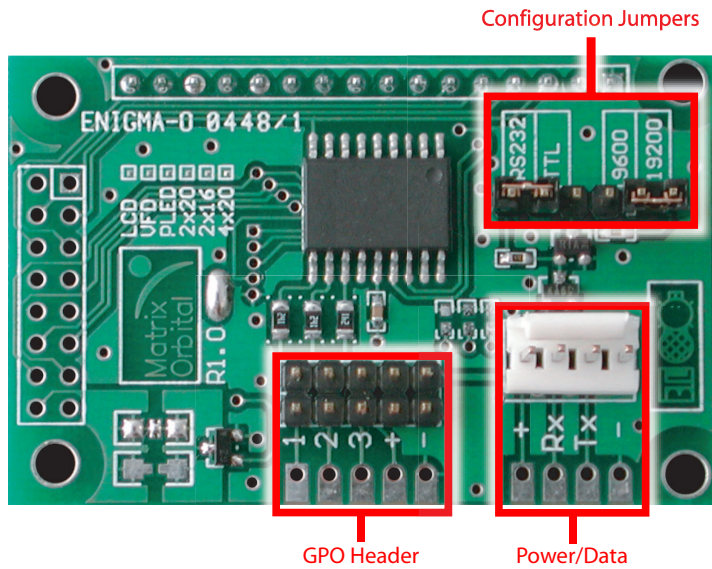


Figure 3: Electrical Connections

The display has four connectors as shown in Table 1.

Table 1: Connectors & Functions

Connector	Function
10 pin dual header	General purpose outputs (3)
4 pin	Power (5.0 VDC), Data (Rx, Tx) and Ground
3 pin header	9600 or 19200 baud selection
3 pin header	RS232 or TTL communication selection



## 2.1.1 Power Connections

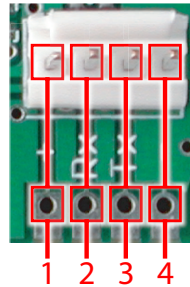
Power is applied via pin1 and ground via pin 4 as shown in Figure 4. Power requirement is +5 VDC  $\pm 0.25V$ . As an alternate power connection, power may also be supplied via the “fingers” located below the 4 pin power connector.

---

### WARNINGS



- Do not apply any power with reversed polarization.
  - Do not apply any voltage other than the specified voltage.
  - Do not use any cables other than the cables supplied by Matrix Orbital, unless aware of the modifications required.
- 



Pin 1	+5.0 VDC
Pin 2	Rx
Pin 3	Tx
Pin 4	Ground

Figure 4: Power Connector

## 2.1.2 Five Volt Modules

If the display is used in a PC it becomes tempting to plug a spare power connector into the unit. **Don't do this!** Wiring for the PC power connector and that required for the display are different as shown in the Figure 5.

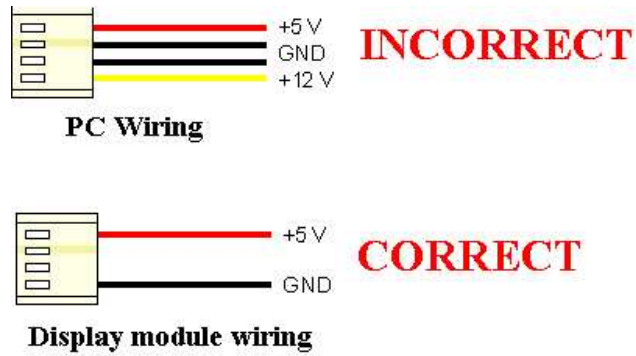


Figure 5: Wiring for 5V Modules



**WARNING** DO NOT under any circumstances plugin an unmodified floppy power cable into the display. This will damage the display and void your warranty.

Matrix Orbital can supply an adapter cable designed for use with the display when it's installed in a PC. The cable is wired as shown in the Figure 6.



Figure 6: 5V Power Cable

## 2.2 Communication Connection

The display communicates at 9600 or 19200 baud on RS-232 or TTL levels. Data from the display is only sent when in TTL mode. The communication protocol is set up as follows: 8 data bits, no parity and 1 stop bit (8N1).

## 2.2.1 RS-232 Communications

This unit is set to RS-232 communication at 19200 baud by factory default. Using the custom cable available for this module, the PC's Rx, Tx and ground connections will be connected to the 4 pin header located on the module. To make a custom for RS-232 communications, refer to Table 2 and Figure 4.

---

### NOTES

- This device complies with the EIA232 standard in that it uses signal levels from +/- 3V to +/- 12V. It will not operate correctly at TTL (0 to +5V) levels without modification. A null modem cable will not work.
  - With RS-232 communications, the display can only receive data, not transmit.
- 

Table 2: DB9 Pinout

Pin Number	Direction	Description	LCD	Host
2	Data from LCD	Data out (LCD)	Tx	Rx
3	Data to LCD	Data in (LCD)	Rx	Tx
5	-	Ground	gnd	gnd

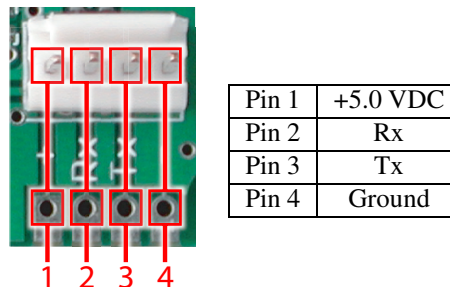


Figure 7: Power/Communication Connector

Please see Figure 8 for default RS-232 communication jumper settings.

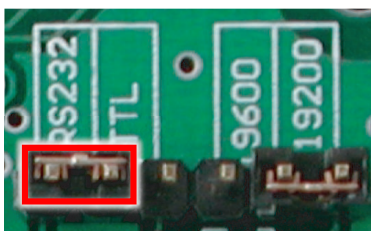


Figure 8: RS-232 Level Selected (Default)

## 2.2.2 TTL Communications

A PC is not to be used when TTL communication is established via a direct com port. Note that this device uses signal levels from 0V to + 5V on the TTL setting. One modification is required for TTL. The communication jumper has to be set to the TTL position. Please see Figure 9 for TTL configuration.

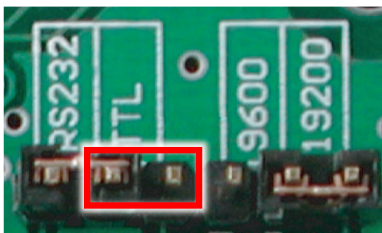
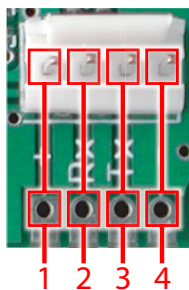


Figure 9: TTL Selected

For TTL communication via the 4 pin connector, please see Figure 4.



Pin 1	+5.0 VDC
Pin 2	Rx
Pin 3	Tx
Pin 4	Ground

Figure 10: Power/Communication Connector

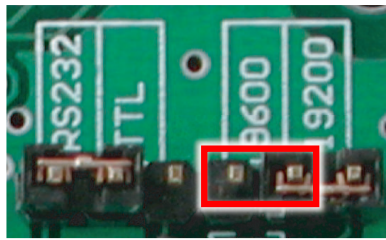
---

**NOTE** Data can be received from the display at TTL levels, such as the module type and firmware version.

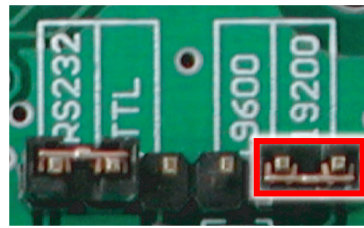
---

## 2.3 Set Baud Rates

With this module you can only select between two baud rates. The baud rates available are 9600 or 19200 baud in 8N1 format. Please see Figure 11 on how to configure for each baud rate.



(a) 9600 Baud



(b) 19200 Baud

Figure 11: Baud Rate Jumper

## 2.4 General Purpose Outputs

The display has three general purpose outputs. These are provided to control relays or other electronic devices. This allows external devices to be turned on or off using the PC or controller and software commands. Please see Figure 13 for GPO pinout.

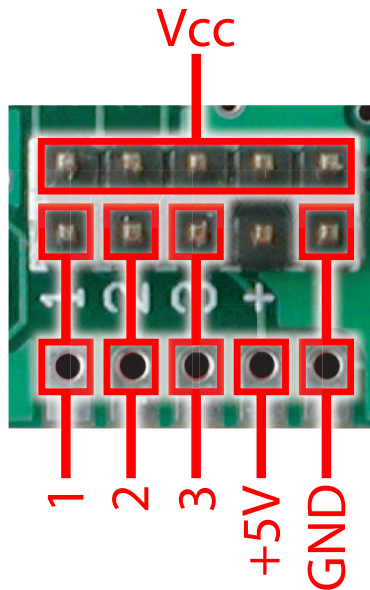


Figure 12: General Purpose Pinout

Each output is wired as shown in Figure 13 . The + terminal is connected directly to the module positive supply, the - terminal is connected through a 240 ohm current limiting resistor and the electronic switch to ground.

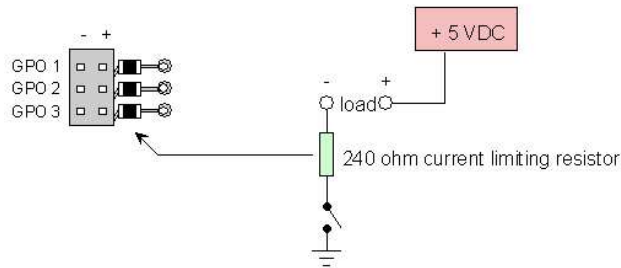


Figure 13: General Purpose Outputs

Maximum allowable current is 20 mA, which is enforced by the current limiting resistor. If the device being switched has a resistance of 240 ohms or more the corresponding resistor may be shorted. To short the corresponding resistor, solder a small jumper wire (wirewrap wire is good) across the resistor(s), or take the resistor(s) out and short the resistor pads.

---

**NOTE** The GPOs do not have any over current or over / under voltage protection so care must be taken when using them. For instance if the external device is a relay it must be fully clamped (using a diode and capacitor) to absorb any generated back electro-motive force (EMF). Please refer to Figure 14 on clamping a relay.

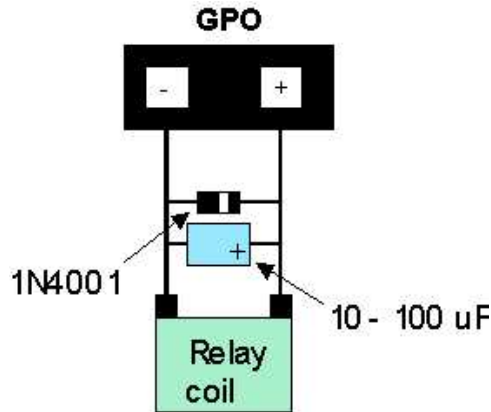


Figure 14: Clamping a Relay

---

### 3 Issuing Commands

Commands are issued to the display by the controller. In a test setup, commands can be issued to the display by means of a Basic program, using the chr\$( ) function. In the tables below, we've shown commands in hex, ASCII and decimal form. All commands begin with the prefix character 0xFE (254 decimal). These commands are issued on the serial communications link (RS-232 or TTL) at the currently defined baud rate.

For example (using BASIC in a test setup), the user could issue the command to clear the screen on the display by including the line:

```
PRINT#1, chr$(254); chr$(88).
```

Or, with C the user could (using Zcomm serial library)

```
ZComm1->WriteCommByte(0xfe); // command prefix hex value  
ZComm1->WriteCommByte(0x58); // command hex value for "clear screen"
```

```
ZComm1->WriteCommByte(0xfe); // command prefix hex value  
ZComm1->WriteCommByte('X'); // command ASCII value for "clear screen"
```

```
ZComm1->WriteCommByte(254); // command prefix decimal value  
ZComm1->WriteCommByte(88); // command decimal value for "clear screen"
```

## 3.1 Issuing Commands From a Basic Stamp

Before issuing commands from a Basic Stamp, the module and the Basic Stamp must be set up to communicate in TTL at 9600 baud. Please see Section 2.3 and Section 2.2.2 for setup configurations.

- Sending text to the display:

```
'{$STAMP BS2}
  SEROUT 1, 84, ["HELLO WORLD"]
```

This will display HELLO WORLD on your display.

SEROUT : is the command to send serial information out

1 : is Serial Port #1

84 : is the speed at which the information is sent at, in this case 9600bps.

- Sending a command to the display:

```
'{$STAMP BS2}
  SEROUT 1, 84, [254]   'Command Prefix
  SEROUT 1, 84, [88]   'Clear screen command
```

In this example, the command to clear the screen is sent to the display. Every command requires a "Command Prefix". With Matrix Orbital displays, that's Hex: FE, Decimal: 254 and ASCII: 254. Properly sent, any format can be used. Decimal format was used in this example.

- Setting the Backlight to go off in 2 minutes:

```
'{$STAMP BS2}
  SEROUT 1, 84, [254]   'Command Prefix
  SEROUT 1, 84, [66]   'Backlight ON command
  SEROUT 1, 84, [2]    'Setting the number of minutes to be on
```

To permanently turn the backlight on, 0 would be sent as the third byte. When the display receives the command, it will know how many more bytes of information it should get. In this 'backlight on' case, the display knows to expect one more byte of information.

- Creating a medium digit:

```
'{$STAMP BS2}
  Digit VAR Byte   'Create the variables Reps VAR NIB
  Digit = 0       'Make sure the variables are 0 Reps = 0
  SEROUT 1, 84, [254]   'Command Prefix
  SEROUT 1, 84, [88]   'Clear screen command
```



```

SEROUT 1, 84, [254]  'Command Prefix
SEROUT 1, 84, [109] 'Initilize Medium Digits command
FOR Reps = 1 TO 10  'A loop to repeat it self 10 times
SEROUT 1, 84, [254]  'Command Prefix
SEROUT 1, 84, [111]  'Display medium digit command
SEROUT 1, 84, [1]   'Display medium digit in row 1
SEROUT 1, 84, [1]   'Display medium digit in column 1
SEROUT 1, 84, [Digit] 'Display the medium digit
Digit = Digit + 1   'Increment by 1
PAUSE 1000         '1 second pause to see the text
NEXT
STOP

```

These 8 custom characters are user defined and can be used however the user sees fit.

---

**NOTE** If Medium Digits are used, Bar Graps or user defined Custom Characters cannot be used at the same time. It has to be one or the other.

---

## 3.2 On Numbers

Like all computerized devices, the display operates with commands and values in the form of binary numbers. These binary numbers are arranged in 8 digit (i.e., 8 bit) groups called bytes. The decimal value of a byte may have any value from 0 to 255.

Bytes are usually specified in either decimal or hexadecimal (base 16) form for convenience, since binary numbers are confusing to deal with directly. Hexadecimal (hex) numbers are particularly convenient because exactly two hexadecimal digits make up one byte, each hex digit representing 4 binary digits (4 bits) as shown in Table 3.

Table 3: Hex Value Table

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Based on Table 3, the byte 01001011 can be represented in hex as 4B, which is usually written as any of 4Bh, 4BH, 4B hex or 0x4B. The numbers can also be expressed in decimal form if preferred.

### 3.3 ASCII Characters

Since computers deal internally with numbers only, but externally with both letters and numbers, several schemes were developed to ‘map’ written characters to numeric values. One such scheme has become universal, the American Standard Code for Information Interchange, or ASCII. ASCII tables are readily available from a number of sources. A few examples are shown in Table 4

Table 4: Example of an ASCII Table

The letter	A	has a value of	65 Decimal or	41 Hex
The letter	a	has a value of	97 Decimal or	61 Hex
The number	0	has a value of	48 Decimal or	30 Hex
The number	9	has a value of	57 Decimal or	39 Hex

This gives rise to the possibility of confusion when parameters are being set on the display. For example, the GPO ON and OFF commands use a number to indicate which GPO is being controlled. We’re told that acceptable values are 1 to 3. **All such parameters must use numeric values (i.e., the actual byte values).** If we send the ASCII number 0 by mistake it will actually give the value 48 decimal (30 hex) to the parameter, which is wrong.

## 3.4 Example Command

Syntax	Hexadecimal	0xFE 0x6F [row] [column] [digit]
	Decimal	254 111 [row] [column] [digit]
	ASCII	254 "o" [row] [column] [digit]

	Parameter	Size	Description
Parameters	row	1	The row number (0 - 2)
	column	1	The column number (1 - 16)
	digit	1	The digit to place (0 - 9)

**Description** Draws a medium [digit] in the specified row and column. Medium digits occupy two rows and a single column. When [row] is specified as '1', the medium digit will be displayed across the first and second rows of the display. When [row] is specified as '2', only the top part of the medium digit will be visible on the second line of the display. When [row] is specified as '1', only the bottom part of the medium digit will be shown on the first line of the display.

Before placing a medium digit, it is advised that you should initialize the custom characters with the "Initialize Medium Digits" command.

**Remembered** No

**Examples** Placing a medium digit on the screen:

```
txUART.sendByte(0xFE); // command prefix
txUART.sendByte('m'); // initialize medium digits
txUART.sendByte(0xFE); // command prefix
txUART.sendByte('o'); // place medium digit command
txUART.sendByte(1); // place the digit on the first and second row
txUART.sendByte(2); // place the digit in the second column
txUART.sendByte(3); // place a '3' on the screen
```

**Syntax** This is a quick summary of the format of a command including the placement of parameters in hexadecimal, decimal and if appropriate ASCII representations.

**Parameters** Each parameter will be outlined along with the length, description and the valid values.

**Description** The description will outline the usage of the command in detail.

**Remembered** The remember command can be a convenient method to set up the start up state of the display. While the remember function is active, changes made by certain commands are written to non-volatile memory. Remembered could be one of three values:

- Yes - any changes made with this command will be remembered as long as the remember function is active.
- No - the state of the remember function does not affect any changes made with this command and they are not committed to non-volatile memory.
- Always - the state of the remember function does not affect any changes made with this command and they are always committed to non-volatile memory.



**WARNING** It is not recommended to leave the remember function active. With it active, it is very easy to reach the maximum write limit of the non-volatile memory which will cause the unit to malfunction.

---

**Examples** If the examples section is present, it will demonstrate the basic usage of the command.

## 4 Text Commands

### 4.1 Introduction

When the display receives a character, it displays that character at the position currently defined. The next character sent to the module then advances to the following position on the display. Characters are drawn using the built in font, and only characters defined in the font are actually displayed. Characters that are not defined by the built in font print as a space (i.e., the cursor is advanced for the next character). The position where text is to be inserted is a character location stored in the display's volatile memory and maintained internally by the display's firmware. The commands in this section perform various functions that involve how text is presented on the display.

To display straight text, send equivalent ASCII, hex or decimal value of the appropriate character. Text is displayed on the MOS-AL-162 using the built in 5x8 dot matrix font. In addition, there are up to 8 user defined characters.

### 4.2 The Built In Character Font

The display includes a built in 5x8 dot matrix font with the full range of ASCII characters plus a variety of extended characters, as shown in the Figure below.

D7	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1			
D6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1			
D5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1			
D4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1			
D3 D2 D1 D0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
0 0 0 0	0				0	0	P	\	P					—	9	≡	0	P	
0 0 0 1	1				!	1	A	0	a	A				μ	7	7	4	ā	ā
0 0 1 0	2				"	2	E	R	l	r				τ	4	9	×	E	0
0 0 1 1	3				#	3	O	S	c	e				┘	0	T	E	E	ε
0 1 0 0	4				\$	4	D	T	d	t				√	I	†	†	M	0
0 1 0 1	5				%	5	E	U	e	u				.	7	†	†	1	0
0 1 1 0	6				&	6	F	V	f	v				ヲ	加	ニ	3	0	Σ
0 1 1 1	7				'	7	G	W	g	w				フ	†	又	7	8	π
1 0 0 0	8				<	8	H	X	h	x				4	0	2	9	J	又
1 0 0 1	9				>	9	I	Y	i	y				e	†	†	†	"	9
1 0 1 0	A				*	:	J	Z	j	z				≡	0	†	†	†	†
1 0 1 1	B				+	:	K	I	k	i				≡	†	†	†	†	†
1 1 0 0	C				,	<	L	†	†	†				≡	†	†	†	†	†
1 1 0 1	D				—	=	N	†	†	†				≡	†	†	†	†	†
1 1 1 0	E				.	>	N	^	n	†				≡	†	†	†	†	†
1 1 1 1	F				/	?	0	_	o	†				≡	†	†	†	†	†

Figure 15: Character Set

## 4.3 Command List

### 4.3.1 Auto Line Wrap On

Syntax	Hexadecimal	0xFE 0x43
	Decimal	254 67
	ASCII	254 “C”
Description	Enables automatic line wrapping. Note that this is not “word wrapping” and wraps may occur in the middle of a word.	

Remembered Yes

### 4.3.2 Auto Line Wrap Off

Syntax           Hexadecimal   0xFE 0x44  
                  Decimal       254 68  
                  ASCII        254 "D"

Description      Disables automatic line wrapping. Characters beyond the end of a line will be lost.

Remembered      Yes

### 4.3.3 Set Cursor Position

Syntax           Hexadecimal   0xFE 0x47 [column] [row]  
                  Decimal       254 71 [column] [row]  
                  ASCII        254 "G" [column] [row]

Parameters	Parameter	Size	Description
	column	1	The column number (1 - 16)
row	1	The row number (1 - 2)	

Description      This command sets the cursor position (text insertion point) to the [column] and [row] specified. Columns have values from 1 to 16 and rows have values from 1 to 2.

Remembered      No

### 4.3.4 Send Cursor Home

Syntax           Hexadecimal   0xFE 0x48  
                  Decimal       254 72  
                  ASCII        254 "H"

Description      This command moves the cursor position (text insertion point) to the top left of the display area.

Remembered      No

### 4.3.5 Turn On Underline Cursor

Syntax	Hexadecimal	0xFE 0x4A
	Decimal	254 74
	ASCII	254 “J”
Description	Turns on the underline cursor. The cursor shows the current text insertion point. Both blinking block and underline cursors may be turned on or off independently. The cursor is off by default.	
Remembered	Yes	

### 4.3.6 Turn Off Underline Cursor

Syntax	Hexadecimal	0xFE 0x4B
	Decimal	254 75
	ASCII	254 “K”
Description	Turns off the underline cursor. Does not affect the underline cursor.	
Remembered	Yes	

### 4.3.7 Cursor Left

Syntax	Hexadecimal	0xFE 0x4C
	Decimal	254 76
	ASCII	254 “L”
Description	Moves the cursor one position to the left but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.	

---

**NOTE** A “destructive backspace”, which erases the character to the left of the original position, may be done by issuing the following sequence: cursor left, space, cursor left.

---

Remembered	No
------------	----

### 4.3.8 Cursor Right

Syntax	Hexadecimal	0xFE 0x4D
	Decimal	254 77
	ASCII	254 “M”
Description	Moves the cursor one position to the right but does not erase any character that may be in that position. Note that this command moves the text insertion point even if the cursor is turned off.	
Remembered	No	

### 4.3.9 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 scroll position (the bottom right character position).	
Remembered	Yes	

### 4.3.10 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. 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	

### 4.3.11 Turn On Blinking Block Cursor



Syntax	Hexadecimal	0xFE 0x53
	Decimal	254 83
	ASCII	254 “S”
Description	Turns on the blinking block cursor. The cursor shows the current text insertion point. Both blinking block and underline cursors may be turned on or off independently. The cursor is off by default.	
Remembered	Yes	

### 4.3.12 Turn Off Blinking Block Cursor

Syntax	Hexadecimal	0xFE 0x54
	Decimal	254 84
	ASCII	254 “T”
Description	Turns off the blinking block cursor. Does not affect the underline cursor.	
Remembered	Yes	

### 4.3.13 Clear Display

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

## 5 Bar Graphs and Special Characters

### 5.1 Introduction

The display includes the ability to draw bar graphs (either horizontal or vertical) and allows users to define up to eight special characters.

Eight characters (ASCII values 0x00 to 0x07) are set aside for use with bar graphs, user defined characters, and large and medium sized numbers. Since the same 8 characters are used for each function, the functions may not be used simultaneously. The characters may be defined or redefined at any time by issuing

commands shown in this section. Once defined, they may be used either by means of the bar graph commands, or by simply issuing one of the ASCII values 0x00 to 0x07, which are not prefixed by the command byte, 254.

## 5.2 Command List

### 5.2.1 Draw Vertical Bar Graph

Syntax	Hexadecimal	0xFE 0x3D [column] [height]	
	Decimal	254 61 [column] [height]	
	ASCII	254 “=” [column] [height]	
Parameters	<b>Parameter</b>	<b>Size</b>	<b>Description</b>
	column	1	The column number (1 - 16)
	height	1	The height value (0 - 16)
Description	<p>Draws a vertical bar graph in [column] having a height of [height] pixels. The height may range from 0 to 16 pixels. The necessary characters must be first initialized by either of the commands Initialize Wide Vertical Bar Graph (254 118), Initialize Narrow Vertical Bar Graph (254 115) or Define Custom Characters (254 78). The said commands will determine the width of the vertical graph drawn. Graphs may be erased by drawing a bar graph of height = 0 in the same column.</p>		
Remembered	No		

### 5.2.2 Load Startup Screen

Syntax	Hexadecimal	0xFE 0x40 [characters]	
	Decimal	254 64 [characters]	
	ASCII	254 “@” [characters]	
Parameters	<b>Parameter</b>	<b>Size</b>	<b>Description</b>
	characters	32	Any character to be defined by the user.

Description This command sets and memorizes the startup screen that will appear each time the display is powered on.

By default the screen shows:

Matrix Orbital MOS-AL-162
------------------------------

The 32 characters define the two 16 character rows of the screen.

If sending more than 10 characters to be stored, add in about 10ms per character delay. Predefined custom characters can also be used in the startup screen, by using 0x00 through 0x07 characters.

Character 1 .....	16
Character 17 .....	32

Remembered Always

### 5.2.3 Define Custom Character

Syntax Hexadecimal 0xFE 0x4E [address] [definiton]  
Decimal 254 78 [address] [definiton]  
ASCII 254 "N" [address] [definiton]

Parameter	Size	Description
address	1	The address (0x00 to 0x07)
definiton	8	The character definition

Description

The display allows up to 8 user defined (custom) characters. These characters occupy the first 8 (0x00 to 0x07) places in the character set. Built-in and custom characters occupy a 5x8 pixel matrix.

A character is defined by issuing the command 254 78 [c] followed by 8 bytes to define each pixel row for the character. [c] is the character number (0x00 to 0x07). The 8 bytes are mapped as shown:

MSB			LSB					
*	*	*	1	2	3	4	5	Data Byte 1
*	*	*	6	7	8	9	10	Data Byte 2
*	*	*	11	12	13	14	15	Data Byte 3
*	*	*	16	17	18	19	20	Data Byte 4
*	*	*	21	22	23	24	25	Data Byte 5
*	*	*	26	27	28	29	30	Data Byte 6
*	*	*	31	32	33	34	35	Data Byte 7
*	*	*	36	37	38	39	40	Data Byte 8

A '1' bit indicates an on (black) pixel, a '0' bit indicates an off (clear) pixel.

Once defined, a character is displayed simply by issuing a value (0x00 to 0x07) corresponding to the character number. The character will be laid out as follows:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
<b>Cursor Line</b>				

---

**NOTE** Custom characters will be erased if any of the "Initialize" commands in this section are issued after defining the custom characters, e.g. Initialize Horizontal Bar Graph, Initialize Medium Digits.

---

Remembered

No

## Examples

### Defining a Degree Symbol:

```
txUart.sendByte(0xFE); // command prefix
txUart.sendByte('N'); // custom character command
txUart.sendByte(0x00); // custom character value 0-7
txUart.sendByte(12); // 8 bytes to create
txUart.sendByte(18); // degree symbol
txUart.sendByte(18);
txUart.sendByte(12);
txUart.sendByte(0);
txUart.sendByte(0);
txUart.sendByte(0);
txUart.sendByte(0);
```

To display the character defined:

```
txUart.sendByte(0x00); // display custom character 0
```

## 5.2.4 Initialize Horizontal Bar Graph

### Syntax

Hexadecimal	0xFE 0x68
Decimal	254 104
ASCII	254 "h"

### Description

This command defines the 8 special / user characters to be blocks suitable for use in drawing horizontal bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of horizontal bar graphs may be drawn unless the characters are re-defined by another command.

### Remembered

No

## 5.2.5 Initialize Medium Digits

### Syntax

Hexadecimal	0xFE 0x6D
Decimal	254 109
ASCII	254 "m"

### Description

This command defines the 8 special / user characters to be lines suitable for use in drawing medium digits. Any previously existing definitions will be lost. Once this command has been issued, any number of medium digits may be drawn unless the characters are re-defined by another command.

Remembered No

## 5.2.6 Draw Medium Digits

Syntax           Hexadecimal   0xFE 0x6F [row] [column] [digit]  
                  Decimal       254 111 [row] [column] [digit]  
                  ASCII        254 "o" [row] [column] [digit]

Parameter	Size	Description
row	1	The row number (0 - 2)
column	1	The column number (1 - 16)
digit	1	Digit (0 - 9)

Description       Draws a medium [digit] in the specified row and column. Medium digits occupy two rows and a single column. When [row] is specified as '1', the medium digit will be displayed across the first and second rows of the display. When [row] is specified as '2', only the top part of the medium digit will be visible on the second line of the display. When [row] is specified as '1', only the bottom part of the medium digit will be shown on the first line of the display.

Before placing a medium digit, it is advised that you should initialize the custom characters with the "Initialize Medium Digits" command.

Remembered No

Examples           Placing a medium digit on the screen:  
`txUart.sendByte(0xFE); // command prefix`  
`txUart.sendByte('m'); // initialize medium digits`  
`txUart.sendByte(0xFE); // command prefix`  
`txUart.sendByte('o'); // place medium digit`  
`txUart.sendByte(1); // place the digit on the first and second rows`  
`txUart.SendByte(2); // place digit in the second column`  
`txUart.sendByte(3); // place a '3' on the screen`

## 5.2.7 Initialize Narrow Vertical Bar Graph

Syntax           Hexadecimal   0xFE 0x73  
                  Decimal       254 115  
                  ASCII        254 "s"

Description	This command defines the 8 special / user characters to be blocks suitable for use in drawing narrow (2 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are re-defined by another command.
Remembered	No

## 5.2.8 Initialize Wide Vertical Bar Graph

Syntax	Hexadecimal 0xFE 0x76 Decimal 254 118 ASCII 254 “v”
Description	This command defines the 8 special / user characters to be blocks suitable for use in drawing wide (5 pixel) vertical bar graphs. Any previously existing definitions will be lost. Once this command has been issued, any number of vertical bar graphs may be drawn unless the characters are re-defined by another command.
Remembered	No

## 5.2.9 Draw Horizontal Bar Graph

Syntax	Hexadecimal 0xFE 0x7C [column] [row] [dir] [length] Decimal 254 124 [column] [row] [dir] [length] ASCII 254 “l” [column] [row] [dir] [length]															
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>column</td> <td>1</td> <td>The column number (1 - 16)</td> </tr> <tr> <td>row</td> <td>1</td> <td>The row number (1 - 2)</td> </tr> <tr> <td>dir</td> <td>1</td> <td>The direction value (0 or 1)</td> </tr> <tr> <td>length</td> <td>1</td> <td>The length (0 - 80)</td> </tr> </tbody> </table>	Parameter	Size	Description	column	1	The column number (1 - 16)	row	1	The row number (1 - 2)	dir	1	The direction value (0 or 1)	length	1	The length (0 - 80)
Parameter	Size	Description														
column	1	The column number (1 - 16)														
row	1	The row number (1 - 2)														
dir	1	The direction value (0 or 1)														
length	1	The length (0 - 80)														
Description	Draws a horizontal bar graph in [row] starting at [column] with a length of [length] pixels. [row] may have a value of 1 or 2, [column] may range from 1 to 16 and length may be from 0 to 80 if the graph can extend the full width of the screen. Each column is 5 pixels wide (spaces between the columns do not count). [dir] specifies the direction: 0 goes from left to right, 1 or any other values go from right to left.															
Remembered	No															

## 5.2.10 Remember Custom Character

Syntax	Hexadecimal	0xFE 0xC2 [address] [definiton]	
	Decimal	254 194 [address] [definiton]	
Parameters	Parameter	Size	Description
	address	1	The address (0x00 to 0x07)
	definiton	8	The character definition
Description	This command will store a custom character. Most common use of this command is to define characters for the startup screen. This command does not affect or alter the current custom characters that have been initialized or defined. The syntax is identical to the command Define Custom Character 254 78.		
Remembered	Always		

# 6 Display Functions

## 6.1 Introduction

The commands listed in this chapter are functions of the display such as contrast, brightness and start up screens.

## 6.2 Command List

### 6.2.1 Backlight On

Syntax	Hexadecimal	0xFE 0x42 [minutes]	
	Decimal	254 66 [minutes]	
	ASCII	254 "B" [minutes]	
Parameters	Parameter	Size	Description
	minutes	1	Backlight value (0 to 90)



Description This command turns the backlight on for [minutes], with [minutes] maximum value of 90. If [minutes] is set to zero (0), the backlight will never turn off. When this command is sent while the Remember function is on, it will be remembered. E.g. This command is sent with a value of 1 for [minutes] and Remember is on. When the power is cycled, the module will remember to turn off backlight after 1 minute.

---

**NOTE** The factory default for backlight is on

---

Remembered Yes

### 6.2.2 Backlight Off

Syntax Hexadecimal 0xFE 0x46  
Decimal 254 70  
ASCII 254 “F”

Description This command turns the backlight off. When this command is sent while the Remember function is on, it will be remembered. When power is cycled, the module will remember to turn off backlight.

Remembered Yes

### 6.2.3 Set Contrast

Syntax Hexadecimal 0xFE 0x50 [contrast]  
Decimal 254 80 [contrast]  
ASCII 254 “P” [contrast]

Parameter	Size	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 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.</p> <p>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 acts like the Set and Save Contrast command 254 145.</p>
Remembered	Yes

## 6.2.4 Set and Save Contrast

Syntax	Hexadecimal	0xFE 0x91 [contrast]						
	Decimal	254 145 [contrast]						
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>contrast</td> <td>1</td> <td>The contrast setting (0 to 255)</td> </tr> </tbody> </table>		Parameter	Size	Description	contrast	1	The contrast setting (0 to 255)
Parameter	Size	Description						
contrast	1	The contrast setting (0 to 255)						
Description	<p>This command works in exactly the same way as the "Set Contrast" command when sent with the Remember function on. When this command is sent, not only does it set the contrast with [contrast] value, but also saves this value in the non-volatile memory so that at power cycle, this setting is restored.</p>							
Remembered	Always							

## 6.2.5 Set and Save Backlight Brightness

Syntax	Hexadecimal	0xFE 0x98 [backlight]						
	Decimal	254 152 [backlight]						
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>backlight</td> <td>1</td> <td>Backlight setting (0 to 255)</td> </tr> </tbody> </table>		Parameter	Size	Description	backlight	1	Backlight setting (0 to 255)
Parameter	Size	Description						
backlight	1	Backlight setting (0 to 255)						
Description	<p>This commands sets and saves [brightness] as default.</p>							

Remembered Always

## 6.2.6 Set Backlight Brightness

Syntax Hexadecimal 0xFE 0x99 [backlight]  
Decimal 254 153 [backlight]

Parameter	Size	Description
backlight	1	Backlight setting (0 to 255)

Description This command sets the backlight settings according to [backlight]

Remembered Yes

# 7 GPO Functions

## 7.1 Introduction

The commands listed in this chapter describe the functionality and control of the general purpose outputs.

## 7.2 Command List

### 7.2.1 General Purpose Output Off

Syntax Hexadecimal 0xFE 0x56 [gpo#]  
Decimal 254 86 [gpo#]  
ASCII 254 "V" [gpo#]

Parameter	Size	Description
gpo#	1	GPO number is from (1 - 3)

Description This command turns OFF any of the general purpose outputs. [gpo#] is from 1 to 3. Note that OFF means that the output floats.

Remembered Yes

### 7.2.2 General Purpose Output On

Syntax	Hexadecimal	0xFE 0x57 [gpo#]						
	Decimal	254 87 [gpo#]						
	ASCII	254 "W" [gpo#]						
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>gpo#</td> <td>1</td> <td>GPO number is from (1 - 3)</td> </tr> </tbody> </table>		Parameter	Size	Description	gpo#	1	GPO number is from (1 - 3)
Parameter	Size	Description						
gpo#	1	GPO number is from (1 - 3)						
Description	This command turns ON any of the general purpose outputs. [gpo#] is from 1 to 3. Note that ON means that the output is pulled low (ground via 240 Ohms).							
Remembered	Yes							

### 7.2.3 Remember GPO

Syntax	Hexadecimal	0xFE 0xC3 [gpo#] [state]									
	Decimal	254 195 [gpo#] [state]									
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>gpo#</td> <td>1</td> <td>GPO number (1 - 3)</td> </tr> <tr> <td>state</td> <td>1</td> <td>state (0 or 1)</td> </tr> </tbody> </table>		Parameter	Size	Description	gpo#	1	GPO number (1 - 3)	state	1	state (0 or 1)
Parameter	Size	Description									
gpo#	1	GPO number (1 - 3)									
state	1	state (0 or 1)									
Description	<p>This command will set the startup state for individual GPOs . When the device is powered up the next time, the GPOs will be set to the values from this command.</p> <p>This command does not affect the current state of the GPOs , only at power up.</p>										
Remembered	Always										

## 8 Communications Section

### 8.1 Introduction

The commands listed in this chapter describe how to configure data flow on the RS232 and TTL port. The display has built in flow control which may be useful when long strings of text are downloaded to the display. Flow control is enabled or disabled by two commands. If flow control is enabled, the display will return an "almost full" message (0xFE) to the controller when its internal buffer fills to a defined level, and an "almost empty" message (0xFF) when the buffer contents drop to a defined level.

## 8.2 Command List

### 8.2.1 Enter Flow Control Mode

Syntax	Hexadecimal	0xFE 0x3A [full] [empty]
	Decimal	254 58 [full] [empty]
	ASCII	254 “:” [full] [empty]

Parameter	Size	Description
full	1	The full byte number (0 to 80)
empty	1	The empty byte number (0 to 80)

**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 80 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 its 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.

**Remembered** No

### 8.2.2 Exit Flow Control Mode

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

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

Remembered No

## 9 Miscellaneous Commands

### 9.1 Introduction

The commands listed in this chapter don't readily fit in any of the other categories.

### 9.2 Command List

#### 9.2.1 Set Serial Number

Syntax Hexadecimal 0xFE 0x34 [serial]  
Decimal 254 52 [serial]  
ASCII 254 "4" [serial]

Parameters	Parameter	Size	Description
	serial	2	The new serial number.

Description Modules may be delivered with the serial number blank. In this case the user may set the desired 2 byte serial number using this one time only command.  
The serial number may be set only once. Any future attempt to execute this command will result to no change; the module will return to the originally set serial number.

Remembered Always

#### 9.2.2 Read Serial Number

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

Description This command will return the serial number of the module as it was previously set. This command will return a 2 byte hexadecimal number. This command is only available in TTL mode.

Remembered No

### 9.2.3 Read Version Number

Syntax           Hexadecimal   0xFE 0x36  
                  Decimal       254 54  
                  ASCII        254 “6”

Description       This command will return the firmware version number of the module.  
                  This command is only available in TTL mode.

Remembered No

### 9.2.4 Read Module Type

Syntax           Hexadecimal   0xFE 0x37  
                  Decimal       254 55  
                  ASCII        254 “7”

Description       This command will return the model type value of the module. This  
                  command is only available in TTL mode. Values for various modules at  
                  the time of this publication are as follows:

LCD0821 - 0x01	LCD2021 - 0x03	LCD2041 - 0x05
LCD4021 - 0x06	LCD4041 - 0x07	LK202-25 - 0x08
LK204-25 - 0x09	LK404-55 - 0x0A	VFD2021 - 0x0B
VFD2041 - 0x0C	VFD4021 - 0x0D	VK202-25 - 0x0E
VK204-25 - 0x0F	GLC12232 - 0x10	GLC24064 - 0x13
GLK24064-25 - 0x15	GLK12232-25 - 0x22	GLK12232-25-SM - 0x24
LK404-AT - 0x31	MOS-AV-162A - 0x32	LK402-12 - 0x33
LK162-12 - 0x34	LK204-25PC - 0x35	LK202-24-USB - 0x36
VK202-24-USB - 0x37	LK204-24-USB - 0x38	VK204-24-USB 0x39
PK162-12 - 0x3A	VK162-12 - 0x3B	MOS-AP-162A - 0x3C
PK202-25 - 0x3D	MOS-AL-162A - 0x3E	MOS-AL-202A - 0x3F
MOS-AV-202A - 0x40	MOS-AP-202A - 0x41	PK202-24-USB - 0x42

Remembered No

### 9.2.5 Remember

Syntax            Hexadecimal    0xFE 0x93 [value]  
                       Decimal            254 147 [value]

Parameter	Size	Description
value	1	Value (0 or 1)

Description      [value] specifies Remember functionality; if set to 0, Remember is off and if set to a non-zero value, Remember is on. This command allows a number of settings, such as cursor state, backlight, etc. to automatically be stored to non-volatile memory so they become new defaults. If any of the following commands are sent and the Remember function is ‘on’, they will be automatically remembered:

- Auto Line Wrap On/Off
- Auto Scroll On/Off
- Backlight On/Off
- Set Backlight Brightness
- Set Contrast
- General Purpose Output On/Off
- Turn On/Off Block (Blinking) Cursor
- Turn On/Off Underline Cursor
- Set Backlight/PLED brightness

In the Command Summary, the above commands are marked with ‘R’ which denotes that Remember function affects the command.

---

**NOTE**

Writing to non-volatile memory is time consuming and slows down the operation of the display.

---



**WARNING**

Non-volatile memory has a ‘write limit’ and may only be changed approximately 100,000 times.

---

Remembered      Always



# 10 Command Summary

## 10.1 Text Commands

Description	Syntax	Page	
Set Cursor Position	Hexadecimal	0xFE 0x47 [column] [row]	18
	Decimal	254 71 [column] [row]	
	ASCII	254 “G” [column] [row]	
Send Cursor Home	Hexadecimal	0xFE 0x48	18
	Decimal	254 72	
	ASCII	254 “H”	
Turn On Underline Cursor	Hexadecimal	0xFE 0x4A	19
	Decimal	254 74	
	ASCII	254 “J”	
Turn Off Underline Cursor	Hexadecimal	0xFE 0x4B	19
	Decimal	254 75	
	ASCII	254 “K”	
Cursor Left	Hexadecimal	0xFE 0x4C	19
	Decimal	254 76	
	ASCII	254 “L”	
Cursor Right	Hexadecimal	0xFE 0x4D	20
	Decimal	254 77	
	ASCII	254 “M”	
Turn On Blinking Block Cursor	Hexadecimal	0xFE 0x53	20
	Decimal	254 83	
	ASCII	254 “S”	
Turn Off Blinking Block Cursor	Hexadecimal	0xFE 0x54	21
	Decimal	254 84	
	ASCII	254 “T”	
Clear Display	Hexadecimal	0xFE 0x58	21
	Decimal	254 88	
	ASCII	254 “X”	
Auto Line Wrap On	Hexadecimal	0xFE 0x43	17
	Decimal	254 67	
	ASCII	254 “C”	

Description	Syntax	Page
Auto Line Wrap Off	Hexadecimal	0xFE 0x44
	Decimal	254 68
	ASCII	254 “D”
Auto Scroll On	Hexadecimal	0xFE 0x51
	Decimal	254 81
	ASCII	254 “Q”
Auto Scroll Off	Hexadecimal	0xFE 0x52
	Decimal	254 82
	ASCII	254 “R”

## 10.2 Bar Graphs and Special Characters

Description	Syntax	Page
Load Startup Screen	Hexadecimal	0xFE 0x40 [characters]
	Decimal	254 64 [characters]
	ASCII	254 “@” [characters]
Initialize Wide Vertical Bar Graph	Hexadecimal	0xFE 0x76
	Decimal	254 118
	ASCII	254 “v”
Initialize Horizontal Bar Graph	Hexadecimal	0xFE 0x68
	Decimal	254 104
	ASCII	254 “h”
Initialize Narrow Vertical Bar Graph	Hexadecimal	0xFE 0x73
	Decimal	254 115
	ASCII	254 “s”
Draw Vertical Bar Graph	Hexadecimal	0xFE 0x3D [column] [height]
	Decimal	254 61 [column] [height]
	ASCII	254 “=” [column] [height]
Draw Horizontal Bar Graph	Hexadecimal	0xFE 0x7C [column] [row] [dir] [length]
	Decimal	254 124 [column] [row] [dir] [length]
	ASCII	254 “ ” [column] [row] [dir] [length]
Initialize Medium Digits	Hexadecimal	0xFE 0x6D
	Decimal	254 109
	ASCII	254 “m”

Description	Syntax	Page	
Draw Medium Digits	Hexadecimal	0xFE 0x6F [row] [column] [digit]	26
	Decimal	254 111 [row] [column] [digit]	
	ASCII	254 “o” [row] [column] [digit]	
Define Custom Character	Hexadecimal	0xFE 0x4E [address] [definiton]	23
	Decimal	254 78 [address] [definiton]	
	ASCII	254 “N” [address] [definiton]	
Remember Custom Character	Hexadecimal	0xFE 0xC2 [address] [definiton]	28
	Decimal	254 194 [address] [definiton]	

### 10.3 Display Functions

Description	Syntax	Page	
Backlight On	Hexadecimal	0xFE 0x42 [minutes]	28
	Decimal	254 66 [minutes]	
	ASCII	254 “B” [minutes]	
Backlight Off	Hexadecimal	0xFE 0x46	29
	Decimal	254 70	
	ASCII	254 “F”	
Set Contrast	Hexadecimal	0xFE 0x50 [contrast]	29
	Decimal	254 80 [contrast]	
	ASCII	254 “P” [contrast]	
Set Backlight Brightness	Hexadecimal	0xFE 0x99 [backlight]	31
	Decimal	254 153 [backlight]	
Set and Save Contrast	Hexadecimal	0xFE 0x91 [contrast]	30
	Decimal	254 145 [contrast]	
Set and Save Backlight Brightness	Hexadecimal	0xFE 0x98 [backlight]	30
	Decimal	254 152 [backlight]	

### 10.4 GPO Functions

Description	Syntax	Page	
General Purpose Output On	Hexadecimal	0xFE 0x57 [gpo#]	31
	Decimal	254 87 [gpo#]	
	ASCII	254 “W” [gpo#]	

Description	Syntax	Page
General Purpose Output Off	Hexadecimal	0xFE 0x56 [gpo#]
	Decimal	254 86 [gpo#]
	ASCII	254 “V” [gpo#]
Remember GPO	Hexadecimal	0xFE 0xC3 [gpo#] [state]
	Decimal	254 195 [gpo#] [state]

## 10.5 Communications Section

Description	Syntax	Page
Enter Flow Control Mode	Hexadecimal	0xFE 0x3A [full] [empty]
	Decimal	254 58 [full] [empty]
	ASCII	254 “:” [full] [empty]
Exit Flow Control Mode	Hexadecimal	0xFE 0x3B
	Decimal	254 59
	ASCII	254 “;”

## 10.6 Miscellaneous Commands

Description	Syntax	Page
Read Serial Number	Hexadecimal	0xFE 0x35
	Decimal	254 53
	ASCII	254 “5”
Read Module Type	Hexadecimal	0xFE 0x37
	Decimal	254 55
	ASCII	254 “7”
Read Version Number	Hexadecimal	0xFE 0x36
	Decimal	254 54
	ASCII	254 “6”
Set Serial Number	Hexadecimal	0xFE 0x34 [serial]
	Decimal	254 52 [serial]
	ASCII	254 “4” [serial]
Remember	Hexadecimal	0xFE 0x93 [value]
	Decimal	254 147 [value]

# 11 Appendix: Specifications and Options

## 11.1 Specifications

Table 50: Environmental Specifications

	Standard Temperature
Operating Temperature	0°C to +50°C
Storage Temperature	-20°C to +70°C
Operating Relative Humidity	90% max non-condensing

Table 51: Electrical Specifications

Supply Voltage	4.75 - 5.25Vdc
Supply Current	9 mA typical
Supply Backlight Current	115 mA typical

Table 52: Optical Characteristics

Number of Characters	32 (16 characters by 2 lines)
Matrix format	5 x 8
Display Area	62.5 x 16.10 mm XxY
Character Size	3.20 x 5.55 mm (XxY), not including underline
Character Pitch	3.7 mm
Line pitch	6.09 mm
Dot Size	0.50 x 0.55 mm (XxY)
Dot Pitch	0.57 x 0.62 mm (XxY)
LED Backlight Life	100,000 hours typical
Color of Illumination	Yellow Green, Ice Blue, Inverse Red, Inverse Yellow and Inverse Blue

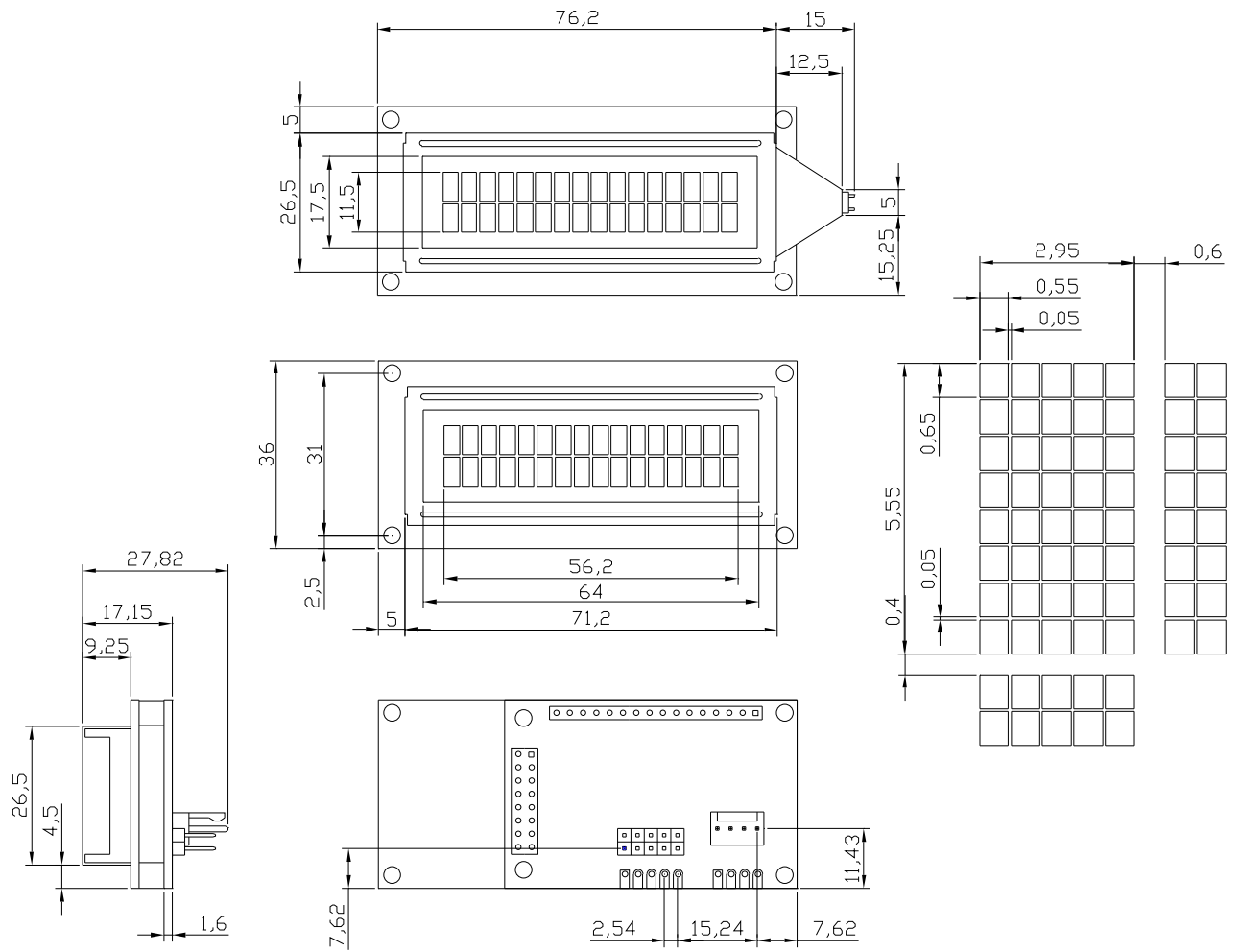


Figure 16: Physical Layout

## 11.2 Options

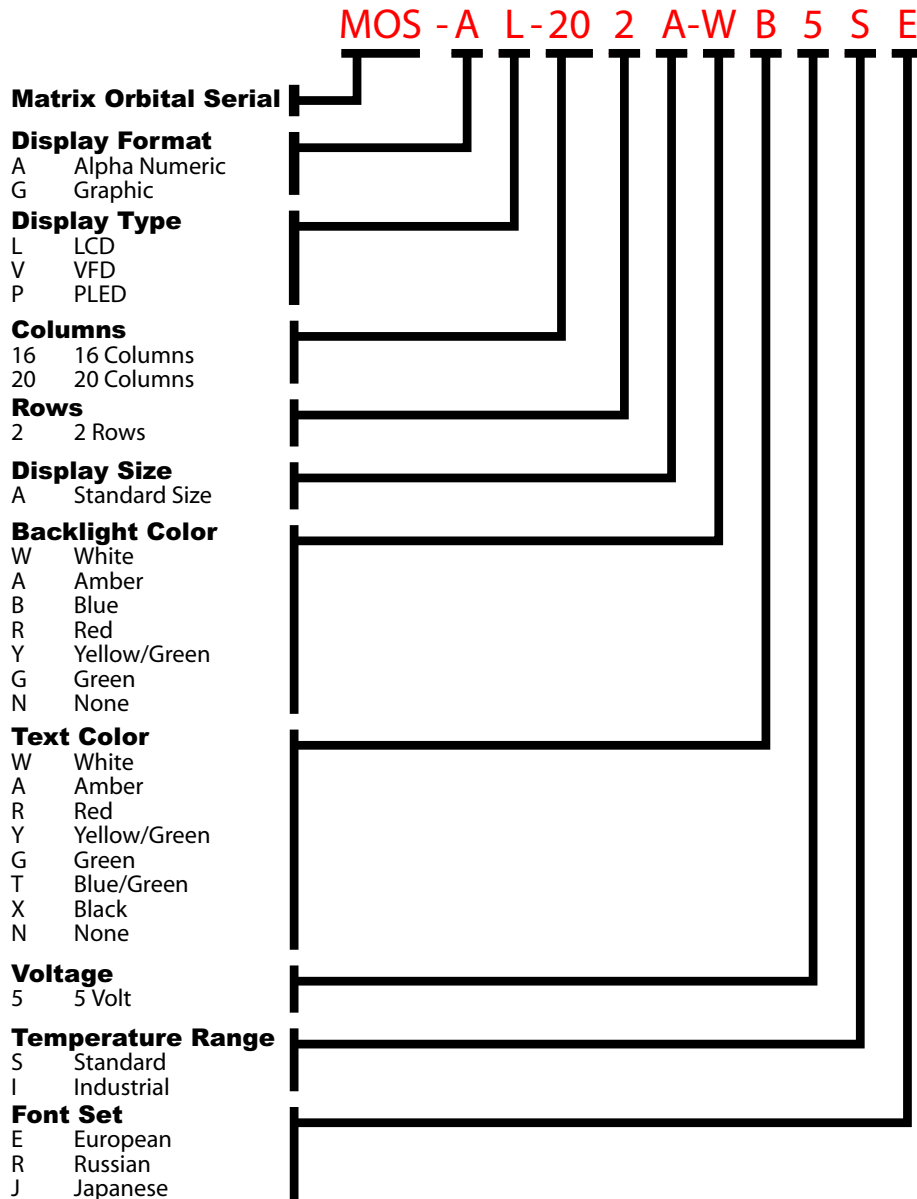


Figure 17: Part Number Classification

## 12 Appendix: Glossary

Table 53: Appendix: Glossary

<b>ASCII</b>	American Standard Code for Information Interchange. A 7 bit binary code representing the English alphabet, decimal numbers and common punctuation marks. Also includes control characters such as carriage return or end of text. An 8 bit superset of the standard ASCII codes is often used today to include foreign characters and other symbols. These supersets are often called extended ASCII character sets.
<b>Backlight</b>	A backlit display is illuminated from behind to provide nighttime and improved daytime readability.
<b>Binary Number</b>	The (data and signaling) bit transmission rate of an RS-232 device.
<b>Bit</b>	A number written using binary notation which only uses zeros and ones.
<b>Bitmap</b>	A representation, consisting of rows and columns of dots, of a graphics image in computer memory. The value of each dot (whether it is filled in or not) is stored in one or more bits of data.
<b>Byte</b>	A grouping of eight binary bits
<b>CCFL</b>	Cold Cathode Fluorescent Lamp. A high brightness backlighting source consists of a fluorescent tube powered by a high voltage A.C. source.
<b>Configuration</b>	The way a system is set up, or the assortment of components that make up the system. Configuration can refer to either hardware or software, or the combination of both.
<b>Contrast</b>	The ratio of luminance between the light state of the display to the dark state of the display.
<b>Controller</b>	The micro-controller or PC used to control the Matrix Orbital display unit.
<b>DB-9</b>	The designation of a connector used in the RS-232 interface: 9 pin connector
<b>Firmware</b>	Software (programs or data) that has been written onto read-only memory (ROM). Firmware is a combination of software and hardware. ROMs, PROMs and EPROMs and flash EEPROMs that have data or programs recorded on them are firmware.



<b>Font</b>	A design for a set of characters. A font is the combination of typeface and other qualities, such as size, pitch, and spacing.
<b>Font Metric</b>	A definition of where font is to be placed, such as margins and spacing between characters and lines.
<b>Hexadecimal</b>	Refers to the base-16 number system, which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F. For example, the decimal number 15 is represented as F in the hexadecimal numbering system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for humans to read hexadecimal numbers than binary numbers.
<b>I<sup>2</sup>C</b>	Short for Inter-IC, a type of bus designed by Phillips Semiconductors in the early 1980s, which is used to connect integrated circuits (ICs). I <sup>2</sup> C is a multi-master bus, which means that multiple chips can be connected to the same bus and each one can act as a master by initiating a data transfer.
<b>Interface</b>	A means by which two systems interact.
<b>LCD</b>	Liquid Crystal Display
<b>Module Type Value</b>	This refers to the model number of the module.
<b>PLED</b>	Polymer Light Emitting Diode. Polymers are substances formed by a chemical reaction in which two or more molecules combine to form larger molecules. PLEDs are thin film displays that are created by sandwiching an undoped conjugated polymer between two proper electrodes at a short distance. The polymer emits light when exposed to electricity.
<b>Pixel</b>	The smallest individually controllable element of a display.
<b>Pre-Generated Fonts</b>	Pre-determined fonts which can be downloaded into graphic liquid crystal displays.
<b>Primitive</b>	A low-level object or operation from which higher-level, more complex objects and operations can be constructed. In graphics, primitives are basic elements, such as lines, curves, and polygons, which you can combine to create more complex graphical images.
<b>RS-232</b>	Short for recommended standard-232C, a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices.
<b>Scroll</b>	To view consecutive lines of data on the display screen. The term scroll means that once the screen is full, each new line appears at the bottom edge of the screen and all other lines move up one position.

<b>Serial Number</b>	A number that is one of a series and is used for identification of the module.
<b>Serial Port</b>	A port, or interface, that can be used for serial communication, in which only 1 bit is transmitted at a time.
<b>Version Number</b>	This refers to the firmware revision number of the module.
<b>VFD</b>	Vacuum Fluorescent Display
<b>Volatile Memory</b>	Temporary memory. Once the power supply is turned off volatile memory is then erased.