MOI-AV202C
Technical Manual

Revision: 1.0

Contents

1 Getting Started

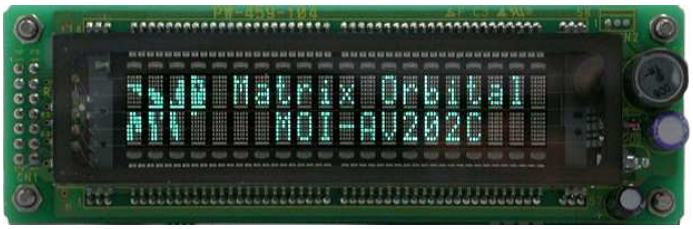


Figure 1: MOI-AV202C

The MOI-AV202C is an intelligent VFD display designed to decrease development time by providing an instant solution to any project. With the ability to communicate via I2C protocol, the versatile MOI-AV202C can be easliy interfaced to any controller. The ease of use is further enhanced by an intuitive command structure to allow display settings such as brightness to be software controlled. 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.. Additionally, up to thirty-two custom characters such as character sets for bar graphs, and medium numbers may be stored in the non-volatile memory to be easily recalled and displayed at any time.

1.1 Display Options Available

The MOI-AV202C affords the ability to add a variety of different colored filters, to allow you to select the display which will best fit your project needs. These options can be found on our e-commerce website at http://www.matrixorbital.com. To contact a sales associate for more information on any of these options, see Section 12.5 for contact information.

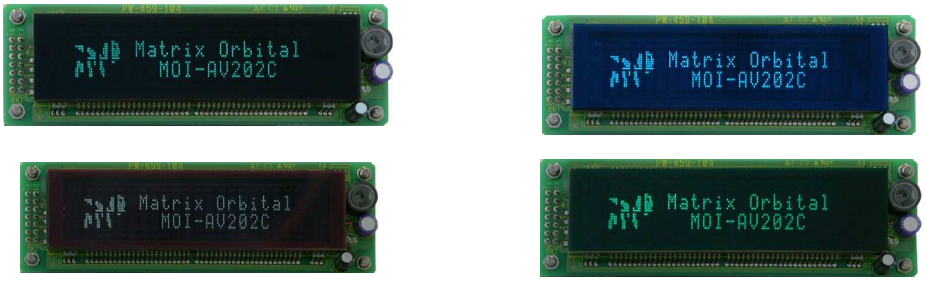


Figure 2: Display Options

1.2 Accessories

Figure 3: Breadboard Cable

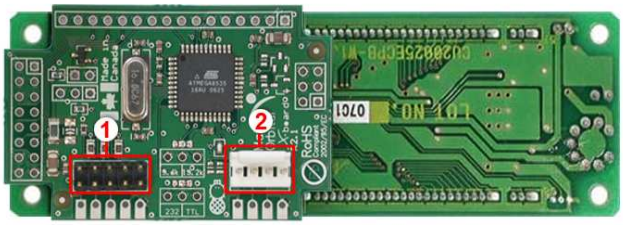
NOTE Matrix Orbital provides all the interface accessories needed to get your display up and running. You will find these accessories and others on our e-commerce website at http://www.matrixorbital.com. To contact a sales associate see Section 12.5 on page 35 for contact information.

1.3 Features

- 20 column by 2 line alphanumeric vacuum florescent display
- I2C communication protocol
- Three, 5V -20mA, general purpose outputs for a variety of applications
- Lightning fast communication, up to 100kHz communication speed
- Use of up to 127 modules on the same 2 wire I2C interface
- Built in font with provision for up to 8 user defined characters
- Four custom character memory banks, which enable storage of up to thrity-two custom characters
- Fully buffered so that no delays in transmission are ever necessary
- Ability to add a customized splash / startup screen
- Horizontal or vertical bar graphs
- Medium digit capability
- Software controlled brightness with configurable time-out setting up to 90 minutes

2 Hardware Information

Refer to the following diagram for this chapter:

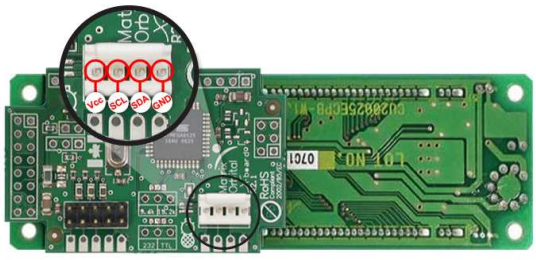


1 Power/Data Connector **2 GPOs**

Figure 4: MOI-AV202C

2.1 Power/Data Connector

The Power/Data Connector provides a standard connector for powering the display module. The MOI-AV202C requires five volts for the standard display module, the voltage is applied through pins one and four of the four pin Power/Data connector. Pins two and three are reserved for the clock and data signals required for I2C transmission.



Pin 1 +5V
Pin 2 SCL
Pin 3 SDA
Pin 4 GND

Figure 5: Alternate Power Connector

Table 1: Power Requirements

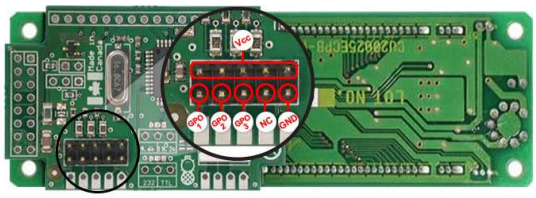
	Standard
Supply Voltage	+5Vdc ±0.25V
Supply Current	198mA
Inrush Current	275mA

WARNINGS

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

2.2 General Purpose Outputs

A unique feature of the MOI-AV202C is the ability to control relays and other external devices using a General Purpose Output, 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 to the above right of the GPOs as pictured below in figure 6. 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.



Vcc +5VDC at 20 mA
GND 0VDC
GPO +5VDC/0VDC

Figure 6: 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.

3 Troubleshooting

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

- First, you will want to make sure that you are supplying power correctly. Measure your power supply output to ensure a steady 5 volt output is available.
- 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 breadboard on the MOI-AV202C. If the breadboard has become loose, or you are unable to resolve the issue, please contact Matrix Orbital, see section 12.5 for contact information.

3.2 The display module is not communicating.

- Check the breadboard cable for continuity. If you don't have an ohm meter, try using a different breadboard cable.
- Ensure that the data is being sent to the correct address. The default slave address for the display module is 0x50.

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

- A common cause may be that the brightness setting has been set to low. The solution to this problem is to change the setting.

4 Communications

4.1 Introduction

The commands listed in this chapter describe how to configure data flow on the MOI-AV202.

4.1.1 I^2C Communication Summary

The MOI-AV202 is capable of communicating at 100 KHz in I^2C mode, with 127 units addressable on a single I^2C communication line. However, in order to communicate via I^2C 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. The MOI-AV202 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 MOI-AV202 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 MOI-AV202, 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 MOI-AV202 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 I^2C the MOI-AV202 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 I^2C 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 MOI-AV202 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 I^2C Transaction Example

The typical I^2C 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: I^2C Transaction Algorithm

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

4.2 Changing the I^2C Slave Address

Syntax
- Hexadecimal: 0xFE 0x33 [adr]
- Decimal: 254 51 [adr]
- ASCII: 254 “3” [adr]

Parameters

Parameter	Length	Description
adr	1	The new I^2C write address (0x00 - 0xFF).

Description

This command sets the I^2C write address of the module between 0x00 and 0xFF. The I^2C write address must be an even number and the read address is automatically set to one higher. For example if the I^2C write address is set to 0x50, then the read address is 0x51.

NOTE The change in address is immediate.

Remembered: Always

Default: 0x50

5 Text

5.1 Introduction

The MOI-AV202 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 formated 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.

5.1.1 Character Set

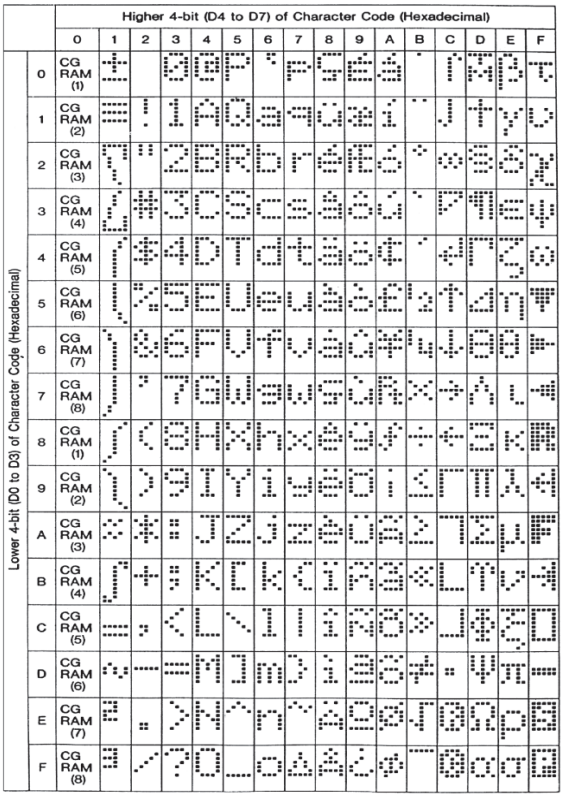


Figure 7: Character Set

5.1.2 Control Characters

In addition to a full text set, the MOI-AV202 display supports the following ASCII Control characters:

0x08 Backspace

0x0C Clear screen / New page

0x0D Carriage return

0x0A Line feed / New line

5.2 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 last row.	
Remembered	Yes	
Default	On	

5.3 Auto Scroll Off

Syntax	Hexadecimal	0xFE 0x52
	Decimal	254 82
	ASCII	254 “R”
Description	When auto scrolling is disabled the text will wrap to the top left corner of the display area when the text reaches the end of last row.	
Remembered	Yes	

5.4 Clear Screen

Syntax	Hexadecimal	0xFE 0x58
	Decimal	254 88
	ASCII	254 “X”
Description	This command will immediately clear all of the contents of the display.	
Remembered	No	

5.5 Changing the Startup Screen

Syntax	Hexadecimal	0xFE 0x40
	Decimal	254 64
	ASCII	254 “@”
Description	In order to change the text that is displayed by the MOI-AV202 when it starts up simply send the command bytes 254 64 followed by the characters that you wish to display, starting from the top left. This command will automatically line wrap the characters that are sent to it.	
Remembered	Yes	

5.6 Set Auto Line Wrap On

Syntax
Hexadecimal	0xFE 0x43
Decimal	254 67
ASCII	254 “C”

Description
Enabling Auto Line Wrap will allow the cursor to automatically wrap over to the next line when the current line is full.

NOTE Line wraps may occur in the middle of a word.

Remembered
Yes

5.7 Set Auto Line Wrap Off

Syntax
Hexadecimal	0xFE 0x44
Decimal	254 68
ASCII	254 “D”

Description
Disabling Auto Line Wrap will allow you to have full control over the cursor position. This means that once the cursor has advanced to the end of a line, it will not wrap over to the next line, unless given the ’Set Cursor Position’ Command.

NOTE All characters written to the display past the end of a line will be lost.

Remembered
Yes

5.8 Set Cursor Position

Syntax
Hexadecimal	0xFE 0x47 [col] [row]
Decimal	254 71 [col] [row]
ASCII	254 “G” [col] [row]

Parameters

Parameter	Length	Description
col	1	Column
row	1	Row

Description This command will allow you to manually set the cursor position, which controls the text insertion point, by specifying the [col] and [row] of the new proposed cursor position.

NOTE If the cursor position is set past the end of a line it will wrap to the beginning of the next line.

Remembered No

5.9 Go Home

Syntax
Hexadecimal	0xFE 0x48
Decimal	254 72
ASCII	254 “H”

Description This command will return the cursor to the top left corner of the display area, identified as row one, column one.

Remembered No

5.10 Move Cursor Back

Syntax
Hexadecimal	0xFE 0x4C
Decimal	254 76
ASCII	254 “L”

Description This command will move the cursor back one space. If this command is sent when the cursor is at the home position the cursor will wrap to the last row / column position if line wrap is on. Sending this command will not effect the text displayed on the module, however any characters that are sent will over write the current characters that are being displayed.

Remembered No

5.11 Move Cursor Forward

Syntax
Hexadecimal	0xFE 0x4D
Decimal	254 77
ASCII	254 “M”

Description This command will move the cursor forward one space. If this command is sent when the cursor is at the bottom right position the cursor will wrap back to the home position if line wrap is on. Sending this command will not effect the text displayed on the module, however any characters that are sent will over write the current characters that are being displayed.

Remembered No

5.12 Underline Cursor On

Syntax
Hexadecimal	0xFE 0x4A
Decimal	254 74
ASCII	254 “J”

Description This command will cause the MOI-AV202 to display an underline cursor at the current text insertion point.

Remembered Yes

5.13 Underline Cursor Off

Syntax
Hexadecimal	0xFE 0x4B
Decimal	254 75
ASCII	254 “K”

Description This command will turn the the underline cursor off.

Remembered Yes

5.14 Blinking Block Cursor On

Syntax
Hexadecimal	0xFE 0x53
Decimal	254 83
ASCII	254 “S”

Description This command will cause the MOI-AV202 to display a block cursor at the current text insertion point.

Remembered Yes

5.15 Blinking Block Cursor Off

Syntax	Hexadecimal	0xFE 0x54
	Decimal	254 84
	ASCII	254 “T”
Description	This command will turn the block cursor off.	
Remembered	Yes	

6 Special Characters

6.1 Introduction

The MOI-AV202 has the ability to create four different sets of eight custom characters and save them to internal banks of memory. Each set of eight can be recalled from memory at any time, and selected characters can be written to the display screen. Characters and sets can be created at any time, saved for later use, and displayed to the screen through the intuitive command structure described below.

6.2 Creating a Custom Character

Syntax	Hexadecimal	0xFE 0x4E [refID] [data]	
	Decimal	254 78 [refID] [data]	
	ASCII	254 “N” [refID] [data]	
Parameters	Parameter	Length	Description
	refID	1	Character reference ID (0-7).
	data	8	Character data.

Description

The MOI-AV202 allows for upta to eight custom defined characters to be added onto the the character set. A custom character is a five by eight pixel matrix with each row represented by a byte value. For example:

Custom Character 'h'					Decimal	Hex
1	*0*	*0*	*0*	*0*	16	0x10
1	*0*	*0*	*0*	*0*	16	0x10
1	*0*	*0*	*0*	*0*	16	0x10
1	*0*	*0*	*0*	*0*	16	0x10
1	*0*	**1**	**1**	*0*	22	0x16
1	**1**	*0*	*0*	**1**	25	0x19
1	*0*	*0*	*0*	**1**	17	0x11
1	*0*	*0*	*0*	**1**	17	0x11

Each bit value of one, in the table, represents an on pixel, whereas a value of zero represents a pixel that is turned off. Therefore in order to define custom character 'h' you would send the command byte prefix 254 followed by the command 78. Next, you will have to select the memory location in which you wish to save the character in. The available memory locations for this command are zero through to seven. After sending the memory location, or [refID], you may then send the eight byte custom character data in sequence from the top to the bottom.

Once you have defined a custom character you may display it by sending the display module the [refID]. For example if a custom character was saved in position one, the command to display the custom character, at the current cursor position, would be simply to send the number one to the display module without quotes.

Remembered

No

6.3 Saving Custom Characters

Syntax

Hexadecimal	0xFE 0xC1 [Bank] [ID] [Data]
Decimal	254 193 [Bank] [ID] [Data]

Parameters

Parameter	Length	Description
Bank	1	Memory bank to save to (0-4).
ID	1	Character ID (0-7)
Data	8	Character Definition

Description

New to the MOI-AV202 has added five non-volatile memory banks for custom character storage. This is intended to allow you to create your own custom bar graphs, medium/large numbers and startup screen. However, each memory bank may be used to store a set of any eight custom characters; with the only provision being that memory bank zero contains the characters that will be used in the startup screen. By default the memory banks will be loaded as follows:

[Bank]	Description
0	Startup screen characters.
1	Horizontal bars
2	Vertical bars
3	Medium numbers

In order to save new custom characters into a memory bank, follow the same process as you would for creating a custom character, see Section 6.2 on page 13, only use 254 193 [Bank Number] before sending the [ID] and character [Data].

Remembered

Yes

6.4 Loading Custom Characters

Syntax

Hexadecimal	0xFE 0xC0 [Bank]
Decimal	254 192 [Bank]

Parameters

Parameter	Length	Description
Bank	1	Memory bank to save to (0-4).

Description

This command is used to load the custom characters into the volatile memory so that they may be used. If custom bar graph or number characters are stored in the memory banks, this command may be used instead of initializing the bar graph / number. To use this command send the command bytes followed by the [Bank] that contains the custom character data that you want to retrieve.

Remembered

No

6.5 Save Startup Screen Custom Characters

Syntax

Hexadecimal	0xFE 0xC2 [refID] [data]
Decimal	254 194 [refID] [data]

Parameters

Parameter	Length	Description
refID	1	Character reference ID (0-7).
data	8	Character data.

Description Using this command you may create the custom characters. that will be stored in memory bank zero, which will be used in the startup screen. For more information about creating custom characters see ***Section 6.2 on page 13***.

NOTES

- Changes only take place once the power has been cycled.
- This command is the same as sending CMD 254 / 193 / 0 / [ID] / [DATA]

Remembered Yes

6.6 Initialize Medium Number

Syntax

Hexadecimal	0xFE 0x6D
Decimal	254 109
ASCII	254 “m”

Description This command will load the default medium number characters into the volatile memory. If you have stored your own custom medium numbers, use the ’Load Custom Characters’ command to load your custom character data into the volatile memory. This command will allow you to use the ’Place Medium Numbers’ command.

Remembered No

6.7 Place Medium Numbers

Syntax

Hexadecimal	0xFE 0x6F [Row] [Col] [Digit]
Decimal	254 111 [Row] [Col] [Digit]
ASCII	254 “o” [Row] [Col] [Digit]

Parameters

Parameter	Length	Description
Row	1	The row number.
Col	1	The column number.
Digit	1	Medium number to place (0-9).

Description This command will place a medium number (two columns high) at the [row] and [col] specified.

NOTE Medium Numbers must be initialized before this command is executed.

Remembered No

6.8 Initialize Horizontal Bar

Syntax	Hexadecimal	0xFE 0x68
	Decimal	254 104
	ASCII	254 “h”

Description: This command will load the default horizontal bar characters into the volatile memory. If you have stored your own custom horizontal bar data, use the ’Load Custom Characters’ command instead to load your custom bar data into the volatile memory. This command will allow you to use the ’Place Horizontal Bar’ command.

Remembered: No

6.9 Place Horizontal Bar Graph

Syntax	Hexadecimal	0xFE 0x7C [Col] [Row] [Dir] [Length]
	Decimal	254 124 [Col] [Row] [Dir] [Length]
	ASCII	254 “|” [Col] [Row] [Dir] [Length]

Parameters:

Parameter	Length	Description
Col	1	The column number.
Row	1	The row number.
Dir	1	The direction of the bar data (0 or 1).
Length	1	The length of the bar data.

Description: This command will place a bar graph at [row], [column]. A [Dir] value of zero will cause the bar to go right, and one will cause the bar to go left. The [Length] is the size in pixels of the bar graph.

NOTES

- Horizontal Bars must be initialized before this command is executed.
- Bar graphs may be one directional only.

Remembered: No

6.10 Initialize Narrow Vertical Bar

Syntax	Hexadecimal	0xFE 0x73
	Decimal	254 115
	ASCII	254 “s”

Description This command will load the narrow vertical bar characters into the volatile memory. If you have stored your own custom vertical bar data, use the 'Load Custom Characters' command instead to load your custom bar data into the volatile memory. This command will allow you to use the 'Place Vertical Bar' command.

NOTE Narrow bars have a width of two pixels.

Remembered No

6.11 Initialize Wide Vertical Bar

Syntax	Hexadecimal	0xFE 0x76
	Decimal	254 118
	ASCII	254 "v"

Description This command will load the wide vertical bar characters into the volatile memory. If you have stored your own custom vertical bar data, use the 'Load Custom Characters' command instead to load your custom bar data into the volatile memory. This command will allow you to use the 'Place Vertical Bar' command.

NOTE Wide bars have a width of five pixels.

Remembered No

6.12 Place Vertical Bar

Syntax	Hexadecimal	0xFE 0x3D [Column] [Length]
	Decimal	254 61 [Column] [Length]
	ASCII	254 "=" [Column] [Length]

Parameters

Parameter	Length	Description
Column	1	The column number.
Length	1	The length of the bar data.

Description This command will place a bar graph at the specified [Column] with the specified [Length]. The [Length] is the size in pixels of the bar graph.

NOTES

- A Vertical Bar style must be initialized before this command is executed.
- Bar graphs may be one directional only.

Remembered No

7 General Purpose Output

7.1 Introduction

General purpose outputs allow you to connect devices, such as LEDs, to the MOI-AV202 and supply them with up to 20mA of current at 5V. The MOI-AV202 has 3 GPOs which are software controlled, with functions to turn them on/off and set the power state for the next startup.

7.2 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

7.3 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 MOI-AV202 output 20mA of current at 5V.

NOTE ON means the output is pulled LOW.

Remembered Yes

7.4 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

8 Display Functions

8.1 Introduction

The MOI-AV202 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.

8.2 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

8.3 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

8.4 Set VFD Brightness

Syntax
Hexadecimal	0xFE 0x59 [brightness]
Decimal	254 89 [brightness]
ASCII	254 “Y” [brightness]

Parameters

Parameter	Length	Description
brightness	1	Brightness setting (0 to 3).

Description This command sets and saves the display’s brightness to [brightness], where [brightness] is a value between 0x00
and 0x03 (between 0 and 3) according to the table below:

Value	Brightness
0x00	0%
0x01	25%
0x01	50%
0x02	75%
0x03	100%

If the remember function is on, this command acts the same as ’Set and Save VFD Brightness’.

Remembered Yes
Default 255

8.5 Set and Save VFD Brightness

Syntax
Hexadecimal 0xFE 0x91 [brightness]
Decimal 254 145 [brightness]

Parameters

Parameter	Length	Description
brightness	1	Brightness setting (0 to 3).

Description
This command sets and saves the display's brightness to [brightness], where [brightness] is a value between 0x00
and 0x03 (between 0 and 3) according to the table below:

Value	Brightness
0x01	25%
0x01	50%
0x02	75%
0x03	100%

Remembered
Always

9 Data Security

9.1 Introduction

Ensuring that your MOI-AV202 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 MOI-AV202 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.

9.2 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

9.3 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 I2C 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	Reserved	Should be left 0
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 Always
Default 0
Examples

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

9.4 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

9.5 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

9.6 Read Customer Data

Syntax		
	Hexadecimal	0xFE 0x35
	Decimal	254 53
	ASCII	254 “5”

Description: Reads whatever was written by Write Customer Data.

Remembered: No

10 Miscellaneous

10.1 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.

10.2 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

10.3 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

Remembered No

11 Command Summary

11.1 Communications

Description	Syntax		Page
Changing the I^2C Slave Address	Hexadecimal	0xFE 0x33 [adr]	7
	Decimal	254 51 [adr]	
	ASCII	254 “3” [adr]	

11.2 Text

Description	Syntax		Page
Auto Scroll On	Hexadecimal	0xFE 0x51	8
	Decimal	254 81	
	ASCII	254 “Q”	
Auto Scroll Off	Hexadecimal	0xFE 0x52	9
	Decimal	254 82	
	ASCII	254 “R”	
Clear Screen	Hexadecimal	0xFE 0x58	9
	Decimal	254 88	
	ASCII	254 “X”	
Changing the Startup Screen	Hexadecimal	0xFE 0x40	9
	Decimal	254 64	
	ASCII	254 “@”	
Set Auto Line Wrap On	Hexadecimal	0xFE 0x43	10
	Decimal	254 67	
	ASCII	254 “C”	
Set Auto Line Wrap Off	Hexadecimal	0xFE 0x44	10
	Decimal	254 68	
	ASCII	254 “D”	
Set Cursor Position	Hexadecimal	0xFE 0x47 [col] [row]	10
	Decimal	254 71 [col] [row]	
	ASCII	254 “G” [col] [row]	
Go Home	Hexadecimal	0xFE 0x48	11
	Decimal	254 72	
	ASCII	254 “H”	
Move Cursor Back	Hexadecimal	0xFE 0x4C	11
	Decimal	254 76	
	ASCII	254 “L”	
Move Cursor Forward	Hexadecimal	0xFE 0x4D	11
	Decimal	254 77	
	ASCII	254 “M”	

Description	Syntax		Page
Underline Cursor On	Hexadecimal	0xFE 0x4A	12
	Decimal	254 74	
	ASCII	254 “J”	
Underline Cursor Off	Hexadecimal	0xFE 0x4B	12
	Decimal	254 75	
	ASCII	254 “K”	
Blinking Block Cursor On	Hexadecimal	0xFE 0x53	12
	Decimal	254 83	
	ASCII	254 “S”	
Blinking Block Cursor Off	Hexadecimal	0xFE 0x54	12
	Decimal	254 84	
	ASCII	254 “T”	

11.3 Special Characters

Description	Syntax		Page
Creating a Custom Character	Hexadecimal	0xFE 0x4E [refID] [data]	13
	Decimal	254 78 [refID] [data]	
	ASCII	254 “N” [refID] [data]	
Saving Custom Characters	Hexadecimal	0xFE 0xC1 [Bank] [ID] [Data]	14
	Decimal	254 193 [Bank] [ID] [Data]	
Loading Custom Characters	Hexadecimal	0xFE 0xC0 [Bank]	15
	Decimal	254 192 [Bank]	
Save Startup Screen Custom Characters	Hexadecimal	0xFE 0xC2 [refID] [data]	15
	Decimal	254 194 [refID] [data]	
Initialize Medium Number	Hexadecimal	0xFE 0x6D	16
	Decimal	254 109	
	ASCII	254 “m”	
Place Medium Numbers	Hexadecimal	0xFE 0x6F [Row] [Col] [Digit]	16
	Decimal	254 111 [Row] [Col] [Digit]	
	ASCII	254 “o” [Row] [Col] [Digit]	
Initialize Horizontal Bar	Hexadecimal	0xFE 0x68	17
	Decimal	254 104	
	ASCII	254 “h”	
Place Horizontal Bar Graph	Hexadecimal	0xFE 0x7C [Col] [Row] [Dir] [Length]	17
	Decimal	254 124 [Col] [Row] [Dir] [Length]	
	ASCII	254 “|” [Col] [Row] [Dir] [Length]	
Initialize Narrow Vertical Bar	Hexadecimal	0xFE 0x73	17
	Decimal	254 115	
	ASCII	254 “s”	
Initialize Wide Vertical Bar	Hexadecimal	0xFE 0x76	18
	Decimal	254 118	
	ASCII	254 “v”	

Description	Syntax		Page
Place Vertical Bar	Hexadecimal	0xFE 0x3D [Column] [Length]	18
	Decimal	254 61 [Column] [Length]	
	ASCII	254 “=” [Column] [Length]	

11.4 General Purpose Output

Description	Syntax		Page
General Purpose Output Off	Hexadecimal	0xFE 0x56 [Num]	19
	Decimal	254 86 [Num]	
	ASCII	254 “V” [Num]	
General Purpose Output On	Hexadecimal	0xFE 0x57 [Num]	19
	Decimal	254 87 [Num]	
	ASCII	254 “W” [Num]	
Set Startup GPO state	Hexadecimal	0xFE 0xC3 [Num] [state]	20
	Decimal	254 195 [Num] [state]	

11.5 Display Functions

Description	Syntax		Page
Display On	Hexadecimal	0xFE 0x42 [min]	20
	Decimal	254 66 [min]	
	ASCII	254 “B” [min]	
Display Off	Hexadecimal	0xFE 0x46	21
	Decimal	254 70	
	ASCII	254 “F”	
Set VFD Brightness	Hexadecimal	0xFE 0x59 [brightness]	21
	Decimal	254 89 [brightness]	
	ASCII	254 “Y” [brightness]	
Set and Save VFD Brightness	Hexadecimal	0xFE 0x91 [brightness]	22
	Decimal	254 145 [brightness]	

11.6 Data Security

Description	Syntax		Page
Set Remember	Hexadecimal	0xFE 0x93 [switch]	22
	Decimal	254 147 [switch]	
Data Lock	Hexadecimal	0xFE 0xCA 0xF5 0xA0 [level]	23
	Decimal	254 202 245 160 [level]	
Set and Save Data Lock	Hexadecimal	0xFE 0xCB 0xF5 0xA0 [level]	24
	Decimal	254 203 245 160 [level]	

Description	Syntax		Page
Write Customer Data	Hexadecimal	0xFE 0x34 [data]	25
	Decimal	254 52 [data]	
	ASCII	254 “4” [data]	
Read Customer Data	Hexadecimal	0xFE 0x35	25
	Decimal	254 53	
	ASCII	254 “5”	

11.7 Miscellaneous

Description	Syntax		Page
Read Version Number	Hexadecimal	0xFE 0x36	25
	Decimal	254 54	
	ASCII	254 “6”	
Read Module Type	Hexadecimal	0xFE 0x37	26
	Decimal	254 55	
	ASCII	254 “7”	

11.8 Command By Number

Command Hex	Description Dec	Page ASCII		
0x33	51	“3”	Changing the I^2C Slave Address	7
0x34	52	“4”	Write Customer Data	25
0x35	53	“5”	Read Customer Data	25
0x36	54	“6”	Read Version Number	25
0x37	55	“7”	Read Module Type	26
0x3D	61	“=”	Place Vertical Bar	18
0x40	64	“@”	Changing the Startup Screen	9
0x42	66	“B”	Display On	20
0x43	67	“C”	Set Auto Line Wrap On	10
0x44	68	“D”	Set Auto Line Wrap Off	10
0x46	70	“F”	Display Off	21
0x47	71	“G”	Set Cursor Position	10
0x48	72	“H”	Go Home	11
0x4A	74	“J”	Underline Cursor On	12
0x4B	75	“K”	Underline Cursor Off	12
0x4C	76	“L”	Move Cursor Back	11
0x4D	77	“M”	Move Cursor Forward	11
0x4E	78	“N”	Creating a Custom Character	13
0x51	81	“Q”	Auto Scroll On	8
0x52	82	“R”	Auto Scroll Off	9
0x53	83	“S”	Blinking Block Cursor On	12

Command Hex	Description Dec	Page ASCII		
0x54	84	“T”	Blinking Block Cursor Off	12
0x56	86	“V”	General Purpose Output Off	19
0x57	87	“W”	General Purpose Output On	19
0x58	88	“X”	Clear Screen	9
0x59	89	“Y”	Set VFD Brightness	21
0x68	104	“h”	Initialize Horizontal Bar	17
0x6D	109	“m”	Initialize Medium Number	16
0x6F	111	“o”	Place Medium Numbers	16
0x73	115	“s”	Initialize Narrow Vertical Bar	17
0x76	118	“v”	Initialize Wide Vertical Bar	18
0x7C	124	“|”	Place Horizontal Bar Graph	17
0x91	145		Set and Save VFD Brightness	22
0x93	147		Set Remember	22
0xC0	192		Loading Custom Characters	15
0xC1	193		Saving Custom Characters	14
0xC2	194		Save Startup Screen Custom Characters	15
0xC3	195		Set Startup GPO state	20

12 Appendix

12.1 Specifications

12.1.1 Environmental

Table 51: Environmental Specifications

Operating Temperature	-40°C to +85°C
Storage Temperature	-50°C to +85°C
Operating Relative Humidity	80% max non-condensing
Vibration (Non-Operating)	10-55-10Hz at 1.0mm for 30min XYZ directions
Shock (Non-Operating)	539 m/s^2 for 10ms
Thermal (Non-Operating)	10°C/min

12.1.2 Electrical

Table 52: Electrical Specifications

Supply Voltage	+5Vdc ±0.25V
Supply Current	198mA typical
Inrush Current	286mA

12.2 Optical Characteristics

Table 53: Optical Characteristics

Character x Lines	20 columns x 2 rows
Module Size	116.00 mm x 37.00 mm x 27.50 mm
Character Size	4.70 mm x 2.40 mm
Active Area	70.80 mm x 11.50 mm

12.3 Physical Layout

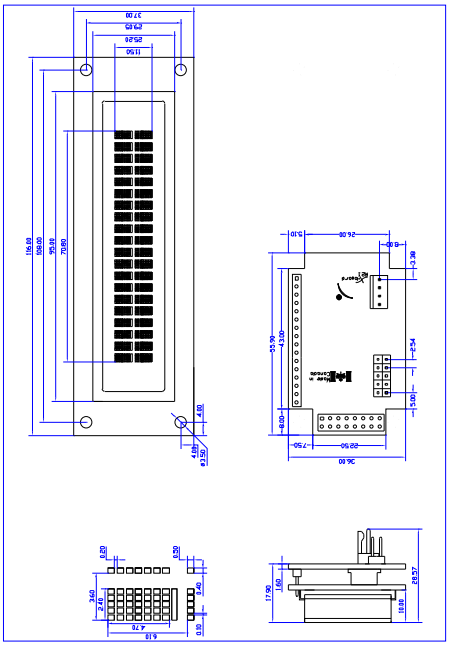


Figure 8: Physical Diagram

Table 54: Revision History

Revision	Description	Author
1.0	Initial Release	Clark

12.4 Definitions

MSB Most Significant Byte

LSB Least Significant Byte

12.5 Contacting Matrix Orbital

Telephone

Sales: 1(403)229-2737

Support: 1(403)204-3750

On The Web

Sales: http://www.MatrixOrbital.com

Support: http://www.MatrixOrbital.ca

Forums: http://www.VFDforums.com

12.6 Revision History