# Economy Series

Including the MOS, MOI, and MOU

## Technical Manual

Revision 1.2

**PCB Revision: 1.0 or Higher**

**Firmware Revision: 5.0 or Higher**

# 1 Revision History

| Revision | Description | Author |
|----------|-------------|--------|
| 1.2 | Added OLED Support | Divino |
| 1.1 | Correction to the GPO Header pin layout | Divino |
| 1.0 | Initial Release | Clark |

# 2 Contents
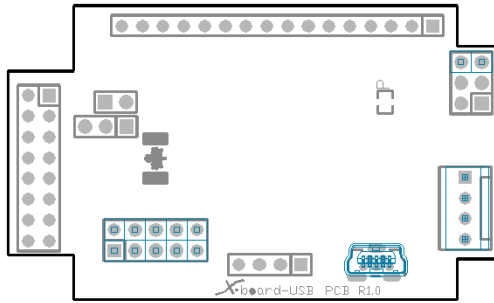
# 3 Introduction



Figure 1: MOU X-Board
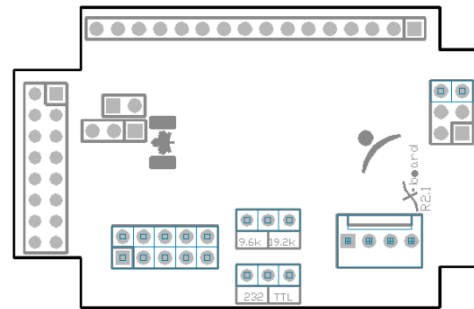


Figure 2: MOS X-Board

The Matrix Orbital Economy Series offers a wide variety of intelligent alphanumeric liquid crystal and vacuum florescent displays designed to decrease development time by providing an instant solution to any project.  At the heart of this line is the elegantly adequate *X*-Board.

Three different versions of the X-Board are available to allow communication to a wide variety of host controllers.  The MOI model offers communication speeds up to 100kbps over a simple two wire, $I^2C$ interface.  The serial MOS version offers 9.6kbps or 19.2kpbs and standard RS232 or TTL logic levels through simple hardware jumpers.  Lastly, the MOU model offers a quick and easy USB connection.

The simple command structure permits easy software control of many settings including backlight brightness, and screen contrast.  On board memory provides up to 40 custom characters which can be saved within the unit and recalled for customized start screens, bar graphs, or larger numbers.  Finally, three general purpose outputs provide simple switchable five volt sources.

The versatile *X*-Board, with all the features mentioned above, is available with a wide colour variety of parallel screen ranging from 8 to 80 characters in size to suit almost any application.

# 4 Quick Connect Guide

## 4.1 Standard Module
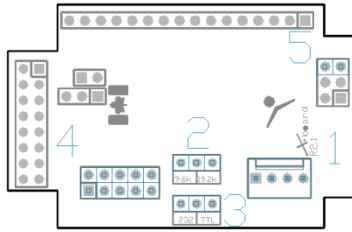


*Figure 3: Standard Connections*

*Table 1: Standard Headers*

| # | Header | Mate |
|---|--------|------|
| 1 | Communication/Power | SCCPC5V/BBC |
| 2 | Protocol Jumper | JMPR |
| 3 | Baud Jumper | JMPR |
| 4 | GPO | None Offered |
| 5 | Manual Override | JMPR |

The serial series *X*-Board allows for user configuration of two common serial protocol logic levels. Either RS232 or TTL levels can be selected using the onboard jumper and accessed through the four pin Communication/Power Header as outlined in the Serial Connections section.

### Recommended Parts



*Figure 4: Communication/Power Cable (SCCPC5V)*

The most common cable choice for any standard Matrix Orbital display, the Communication/Power Cable offers a simple connection to the unit with familiar interfaces. DB9 and floppy power headers provide all necessary input to drive your display.



*Figure 5: Breadboard Cable (BBC)*

For a more flexible interface to the *X*-Board, a Breadboard Cable may be used. This provides a simple four wire connection that is popular among developers for its ease of use in a breadboard environment.

## 4.2 Serial Connections

Serial protocol provides a classic connection to the *X*-Board.  The Communication/Power Cable is most commonly used for this set up as it provides connections for DB9 serial and floppy power cables.  To place your board in Serial mode, adhere to the steps laid out below.

1.  Set the Protocol Select jumper.
    *   RS232: Connect the leftmost, outside RS232 pin to the inside pin of the protocol block.
    *    TTL: Connect the rightmost, outside TTL pin to the centre protocol pin.
2.  Set the Baud Rate jumper.
    *   9600bps: Connect the outside pin on the left, labelled 9.6K to the centre post of the baud block.
    *   19200bps: Connect the 19.2k pin on the outside right to the inside baud pin.
3.  Make the connections.
    a.  Connect the four pin female header of the Communication/Power Cable to the Communication/Power Header of your *X*-Board.
    b.  Insert the male end of your serial cable to the corresponding DB9 header of the Communication/Power Cable and the mate the female connector with the desired communication port of your computer.
    c.  Select an unmodified floppy cable from a PC power supply and connect it to the power header of the Communication/Power Cable.
4.  Create.
    *   uProject or hyperterminal will serve to get you started, then move on with your own development.  Instructions for the former can be found below and a variety of application notes are available for the latter at www.matrixorbital.ca/appnotes.

## 4.3 I$^2$C Module
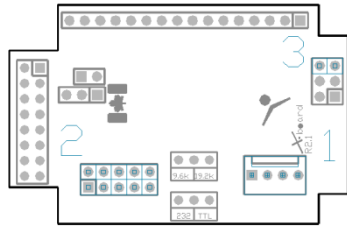


*Figure 6: Standard Connections*

| # | Header | Mate |
|---|--------|------|
| 1 | Communication/Power | SCCPC5V/BBC |
| 2 | GPO | None Offered |
| 3 | Manual Override | JMPR |

The I$^2$C version of the *X*-Board offers a simple application of the two wire I$^2$C interface.  This allows the unit to become one of many slave devices driven by an I$^2$C master device.  This protocol can also be accessed through the four pin Communication/Power Header as outlined in the I$^2$C Connections section.

### Recommended Parts



*Figure 7: Breadboard Cable (BBC)*

For a more flexible interface to the *X*-Board, a Breadboard Cable may be used.  This provides a simple four wire connection that is popular among developers for its ease of use in a breadboard environment.

## I²C Connections

A more advanced connection to the *X*-Board is provided by the I²C protocol setting. It requires a connection directly to your project, best accomplished using a breadboard and the included cable of the same name. Power must be supplied from your breadboard or another external source. To dive right into your application with the I²C model *X*-Board, get started with the guidelines below.

1. Set the Protocol.
   - I²C: The default write address of the *X*-Board is 80, hex 50, while read is one higher at 81. Please ensure the host device uses this address when initially communicating to the board.
2. Make the connections.
   a. Connect the Breadboard Cable to the Communication/Power Header on your *X*-Board and plug the four leads into your breadboard.  The red lead will require power, while the black should be connected to ground, and the green and yellow should be connected to your controller clock and data lines respectively.
   b. Pull up the clock and data lines to five volts using a resistance between one and ten kilohms on your breadboard.
3. Create.
   - This time you're on your own.  While there are many examples within the Matrix Orbital AppNote section, www.matrixorbital.ca/appnotes, too many controllers and languages exist to cover them all.  If you get stuck in development, it is possible to switch over to another protocol on the standard board, and fellow developers are always on our forums for additional support.

## 4.4 USB Module
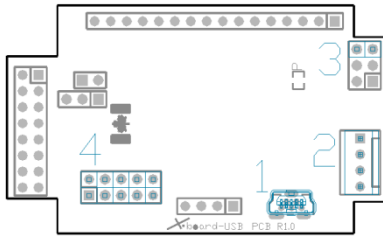


*Figure 8: USB Connections*

*Table 3: USB Headers*

| # | Header | Mate |
|---|--------|------|
| 1 | USB Connector | EXTMUSB3FT |
| 2 | Alternate Power Header | PCS |
| 3 | GPO | None Offered |
| 3 | Manual Override | JMPR |

The USB version of the *X*-Board offers a single USB protocol for easy connection to a host computer. The simple and widely available protocol can be accessed using the on board mini B style USB connector as outlined in the USB Connections section.

### Recommended Parts



*Figure 9: Mini USB Cable (EXTMUSB3FT)*

The External Mini USB Cable is recommended for the USB model of the *X*-Board.  It will connect to the miniB style header on the unit and provide a connection to a regular A style USB connector, commonly found on a PC.

## USB Connections

The USB connection is the quickest, easiest solution for PC development.  After driver installation, the *X-Board* will be accessible through a virtual serial port, providing the same result as a serial setup without the cable hassle.  To connect to your USB series *X*-Board, please follow the steps below.

1. Set the Protocol.
   - USB: The USB version of the *X*-Board offers USB protocol only, no changes are required.
2. Make the connections.
   - Plug the mini-B header of your external mini USB cable into your *X*-Board and the regular USB header into your computer USB jack.
3. Install the drivers.
   a. Download the latest drivers at www.matrixorbital.ca/drivers, and save them to a known location.
   b. When prompted, install the USB bus controller driver automatically
   c. If asked, continue anyway, even though the driver is not signed
   d. When the driver install is complete, your display will turn on, but communication will not yet be possible.
   e. At the second driver prompt, install the serial port driver automatically
   f. Again, if asked, continue anyway
4. Create.
   - Use uProject or hyperterminal to get started, and then move on with your own development. Instructions for the former can be found below and a number of application notes are available for the latter at www.matrixorbital.ca/appnotes.

# 5 Software

The multiple communication protocols available and simple command structure of the *X*-Board means that a variety of applications can be used to communicate with the display.  Text is sent to the display as a character string, for example, sending the decimal value 41 will result in an 'A' appearing on the screen.  A number of control characters are also activated.  Commands are merely values prefixed with a special command byte, 254 in decimal.  While many software programs are available to communicate with the *X*-Board, a number of more common samples are detailed in depth below.

*Table 4: Reserved Control Characters*

| Control Characters | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | Backspace | 10 | Line feed / New line | 12 | Clear screen / New page | 13 | Carriage return |

## 5.1 Hyperterminal

Installed on most Windows computers, hyperterminal can be run by selecting run and typing 'hypertrm' in the command line.  This basic program will allow communication between a PC and your display.

When starting up, a name must be given to your connection, and an icon may be chosen, neither is consequential.  Next, it's important to select the appropriate communication port to which your display is connected.  Finally, the settings below must be entered to complete the port setup.

*Table 5: Hyperterminal Settings*

| BPS | Data Bits | Parity | Stop Bits | Flow Control |
|---|---|---|---|---|
| 19200* | 8 | None | 1 | None |

**Note:** 9600bps Baud may also be selected on the standard *X*-Board model.

Once a port is successfully set up, data can be sent to an attached display by typing on the keyboard.  At this point, it may be helpful to echo keys to the monitor by selecting properties from the file menu and opening the ASCII settings from settings tab.

Commands can be sent to an attached display by issuing decimal commands using the number pad.  While the ALT key is held down, four digit decimal values can be sent as a single ASCII character.  For example, to clear the screen, try the following sequence.

**ALT +0254 ALT +0088**

*Figure 10: Hyperterminal Command*

Any commands or text desired can be sent to the communication port using this method to provide total control of any Matrix Orbital display.

## 5.2 uProject

The Matrix Orbital alphanumeric display tuner, or uProject, is offered as a free download from the www.matrixorbital.ca support site. It allows the basic functionality of any display* to be tested using a simple graphical user interface system.

While basic functionality can be tested using the GUI portion of the program, more advanced users will enjoy the scripting capability found in the uploader tab. Here commands can be stacked, run, and saved for later use. Although many commands are available to be dragged into the script dialog, perhaps the most powerful is the raw data command found in the other branch.

This command allows raw bytes to be sent to the display, permitting many different formats for entry and displaying in decimal notation. Any command from this manual may be entered in decimal notation separated by slashes.

**/254/ /88/**

*Figure 11: uProject Command*

Again, the clear screen command is sent to a connected display, this time using uProject raw data command style. Scripts can be run as a whole using the execute command from the script menu, or as single commands by selecting execute once. Before issuing commands, it is a good idea to ensure communication with a display is successful using some of the basic GUI functions in the main window.

This program provides scratch pad upon which a tome of display projects and ideas can be assembled.

**\*Note:** The uProject AutoDetect function will not perform correctly when a USB display is connected. Please manually configure any USB display.

## 5.3 Application Notes

Full demonstration programs and code are available for Matrix Orbital displays in the C# language from Simple C# AppNote Pack in the Matrix Orbital Application Note section at www.matrixorbital.ca/appnotes. Difficulty increases from beginner, with the Hello World program, to advanced with the Dallas One-Wire temperature reading application.

Many additional applications are available in a number of different programming languages. These programs are meant to showcase the capability of the display and are not intended to be integrated into a final design. For additional information regarding code, please read the On Code document also found on the support site.

# 6 Hardware

## 6.1 Standard Model

### Communication/Power Header
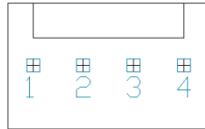


Figure 12: Communication/Power Header

Table 6: Communication/Power Pinout

| Pin | Function |
|-----|----------|
| 1 | Vcc |
| 2 | Rx |
| 3 | Tx |
| 4 | Gnd |

The Communication/Power Header provides a standard connector for interfacing to the serial series *X*-Board. Voltage is applied through pins one and four of the four pin Power/Data connector. Please ensure the correct voltage input for your display by referencing the electrical specifications in Table 20 before connecting power. Pins two and three are reserved for serial transmission, using either RS-232 or TTL logic levels, depending on what has been selected by the Protocol Select Jumpers. The versatile Tyco 640456-4-LF style header employed here can be mated to a wide array of female connectors, a Molex 22-01-3047 for example, for a perfect fit in any project.

### Protocol Select Jumpers

The Protocol Select Jumpers, provide the means necessary to toggle the serial *X*-Board model between RS-232 and TTL logic levels as well as the two baud rate speeds offered. As a default, the jumpers are left unset. In order to place the display module in RS232 mode, connect the two leftmost pins with the jumper provided, for TTL use the rightmost. Similarly, in order to set the display speed, connect the rightmost pins for 19200bps or the leftmost for a more leisurely 9600bps. Jumper tables are shown below with "X" indicating a connection for quick reference.

Table 7: Protocol Settings

| RS232 | | TTL | |
|---|---|---|---|
| X | X | O | O | X | X |

Table 8: Baud Rate Settings

| 9600 | | 19200 | |
|---|---|---|---|
| X | X | O | O | X | X |

## 6.2 I$^2$C Model

### Communication/Power Header
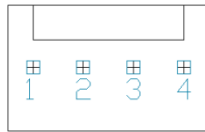


Figure 13: Communication/Power Header

Table 9: Communication/Power Pinout

| Pin | Function |
|-----|----------|
| 1   | Vcc      |
| 2   | SCL      |
| 3   | SDA      |
| 4   | Gnd      |

The now familiar Communication/Power Header also provides a standard connector for interfacing to the *X*-Board I$^2$C model.  Again, voltage is applied through pins one and four of the four pin connector.  Please ensure the correct voltage input for your display by referencing the electrical specifications in Table 20 before connecting power.  Pins two and three are reserved here for I$^2$C communication on addresses 80 and 81.  Proving its versatility, the Tyco 640456-4-LF style header employed again for this connection can be mated to a wide array of female connectors for a perfect fit in any project.

## 6.3 USB Model

### Mini USB Connector
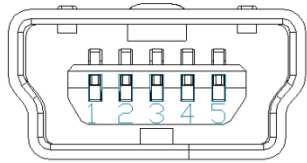


Figure 14: Mini USB Connector

*Table 10: Mini USB Pinout*

| Pin | Function |
|-----|----------|
| 1 | Vcc |
| 2 | D- |
| 3 | D+ |
| 5 | Gnd |

The *X*-Board USB model comes with a familiar Mini USB Connector to fulfill both communication and power needs.  The standard MiniB style header can be connected to any other USB style using the appropriate cable.  Most commonly used with a PC, this connection creates a virtual com port that offers a simple power solution with a familiar communication scheme.

### Alternate Power Connector
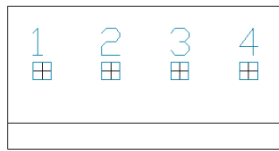


Figure 15: Alternate Power Connector

*Table 11: Alternate Power Pinout*

| Pin | Function |
|-----|----------|
| 1 | NC |
| 2 | Gnd |
| 3 | Gnd |
| 4 | Vcc |

The Alternate Power Connector provides the ability to power the *X*-Board using a *second cable.  The Tyco 171825-4 style header is particularly useful for connecting to an unmodified floppy power cable, a 171822-4 for example, from a PC power supply for a simple bench power solution.

**\*Note:** If the X-Board is powered by the Alternate Power Connector, the 0 ohm jumper resistor near the top right labelled 'P' should be removed to avoid power contention.

## 6.4 Common Features

### General Purpose Outputs
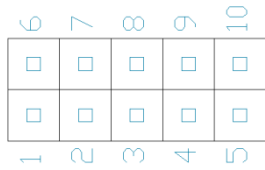


*Figure 16: GPO Header*

*Table 12: GPO Pinout*

| Pin | Function | Pin | Function |
|-----|----------|-----|----------|
| 1 | GPO 1 | 6 | VCC |
| 2 | GPO 2 | 7 | VCC |
| 3 | GPO 3 | 8 | VCC |
| 4 | NC | 9 | VCC |
| 5 | GND | 10 | VCC |

A unique feature of the *X*-Board is the ability to control relays* and other external devices using one of three General Purpose Outputs.  Each can provide up to 20 mA of current at five volts.  The two row, ten pin header can be interfaced to a number of female connectors to provide control to any peripheral devices required.

**Note:** 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.

# 7 Troubleshooting

## 7.1 Power

In order for your Matrix Orbital display to function correctly, it must be supplied with the appropriate power. If the screen does not illuminate, power may not applied correctly. Try following the tips below.

- First, make sure that you are using the correct power connector. Standard floppy drive power cables from your PC power supply may fit on the Communication/Power Header; however they do not have the correct pin out to provide power. Matrix Orbital supplies power cable adapters for connecting to a PC, which can be found in the accessories section.
- 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 interface connector in use on your display. If the power connections have become loose, or you are unable to resolve the issue, please contact Matrix Orbital for more information.

## 7.2 Display

If your display is powered successfully, the Matrix Orbital logo, or user created screen should display on start up. If this is not the case, check out these tips.

- Ensure the contrast is not too high or too low. This can result in a darkened or blank screen respectively. See the Manual Override section to reset to default.
- Make sure that the start screen is not blank. It is possible to overwrite the Matrix Orbital logo start screen, if this happens the screen may be blank. Try writing to the display to ensure it is functional, after checking the contrast above.

## 7.3 OLED Burn-In

OLED technology is susceptible to a burn-in effect; whereby the brightness of active pixels may differ from inactive ones over an extended period of activation. To reduce the burn-in effect, please use a dynamic screen saver or simply turn the screen off when not in active use.

## 7.4 Communication

When communication of either text or commands is interrupted, try the steps below.

- First, check the communication cable for continuity.  If you don't have an ohm meter, try using a different communication cable.  If you are using a PC try using a different Com Port.
- Next, please ensure that the display module is set to communicate on the protocol that you are using, by checking the Protocol Select Jumpers.
- In serial protocol, ensure that the host system and display module are both communicating on the same baud rate.  The display will communicate at either 9600bps or 19200bps only.
- Match Rx from the *X*-Board to the transmitting pin from your host and the Tx pin to the receiving pin.
- If you are communicating to the display via I²C**\*** please ensure that the data is being sent to the correct address.  The default slave address for the display module is 80.
- In I²C mode, connect Rx to the data line of your controller and Tx to the data output.
- Unlock the display.  See the Set and Save Data Lock command for more info.
- Finally, you may reset the display to its default settings using the Manual Override procedure outlined below.

**\*Note:** I²C communication will always require pull up resistors on SCL and SDA of one to ten kilohms.

## 7.5 Manual Override

Should the settings of your display become altered in a way that dramatically impacts usability, the default settings can be temporarily restored.  To override the display, please follow the steps below.

1. Disconnect power from your display.
2. Place a jumper on the override pins.
3. Reconnect power to your unit, and wait for the start screen before removing the override jumper.
4. Settings will be temporarily**\*\*** overridden to the defaults listed in the Manual Override Settings table.  At this point any important settings, such as contrast, backlight, or baud rate, should not only be set but saved so they remain when the override is removed.

| Parameter | Value |
|---|---|
| Backlight | 255 |
| Contrast | 128 |
| I²C Address | 0x50 |

*Table 13: Manual Override Settings*

**\*\*Note:** The display module will revert back to the old settings once turned off, unless desired settings are saved.

# 8 Commands

## Communications

| 1.1. Changing the I2C Slave Address | Dec | **254 51** | **Address** |
|---|---|---|---|
| | Hex | **FE 33** | **Address** |
| | ASCII | **■ 3** | **Address** |

Immediately changes the I2C write address.  Only even values are permitted as the next odd address will become the read address.

| Address | 1 byte,  even value |
|---|---|

## Text

| 1.2. Auto Scroll On | Dec | **254 81** |
|---|---|---|
| | Hex | **FE 51** |
| | ASCII | **■ Q** |

The entire contents of screen are shifted up one line when the end of the screen is reached.  Display default is auto scroll on.

| 1.3. Auto Scroll Off | Dec | **254 82** |
|---|---|---|
| | Hex | **FE 52** |
| | ASCII | **■ R** |

New text is written over the top line when the end of the screen is reached.  Display default is auto scroll on.

| 1.4. Clear Screen | Dec | **254 88** |
|---|---|---|
| | Hex | **FE 58** |
| | ASCII | **■X** |

Clears the contents of the screen.

| 1.5. Changing the Start Up Screen | Dec | **254 64** | **Characters** |
|---|---|---|---|
| | Hex | **FE 40** | **Characters** |
| | ASCII | **■ @** | **Characters** |

Changes the message displayed on start up.  Custom characters can be included by adding their decimal value (0-7).  Characters will automatically wrap on the display.

| Characters | 80 bytes, space characters can be added as needed |
|---|---|

| 1.6. Set Auto Line Wrap On | Dec | **254 67** |
|---|---|---|
| | Hex | **FE 43** |
| | ASCII | **■ C** |

Text will wrap to the next consecutive line once a row becomes full.

| 1.7. Set Auto Line Wrap Off | Dec | **254 68** |
|---|---|---|
| | Hex | **FE 44** |
| | ASCII | **■ D** |

Text will skip one line when wrapping once a row becomes full.  Writing order will be rows 1, 3, 2, and then 4.

| 1.8. Set Cursor Position | Dec | **254 71** | **Column Row** |
|---|---|---|---|
| | Hex | **FE 47** | **Column Row** |
| | ASCII | **■ G** | **Column Row** |

Sets the cursor to s specific position where the next transmitted character is printed.

| Column | 1 byte, value between 1 and display width |
|---|---|
| Row | 1 byte, value between 1 and display height |

| 1.9. Go Home | Dec | **254 72** |
|---|---|---|
| | Hex | **FE 48** |
| | ASCII | **■ H** |

Returns the cursor to the top left of the screen.

| 1.10. Move Cursor Back | Dec | **254 76** |
|---|---|---|
| | Hex | **FE 4C** |
| | ASCII | **■ L** |

Moves cursor one position to the left.  Cursor will obey wrap settings.

| 1.11. Move Cursor Forward | Dec | **254 77** |
|---|---|---|
| | Hex | **FE 4D** |
| | ASCII | **■ M** |

Moves cursor one position to the right.  Cursor will obey wrap settings.

| 1.12. Underline Cursor On | Dec | **254 74** |
|---|---|---|
| | Hex | **FE 4A** |
| | ASCII | **■ J** |

Displays a line under the current cursor position.  Can be used with block cursor.

| 1.13. Underline Cursor Off | Dec | **254 75** |
|---|---|---|
| | Hex | **FE 4B** |
| | ASCII | **■ K** |

Removes line under current cursor position.

| 1.14. Blinking Block Cursor On | Dec | **254 83** |
|---|---|---|
| | Hex | **FE 53** |
| | ASCII | **■ S** |

For VFD and LCD modules a blinking block will be displayed over the current cursor position. On OLED modules the last character printed will blink. Can be used with underline.

| 1.15. Blinking Block Cursor Off | Dec | **254 84** |
|---|---|---|
| | Hex | **FE 54** |
| | ASCII | **■ T** |

Removes blinking block over current cursor position.

## Special Characters

| 1.16. Creating a Custom Character | Dec | **254 78** | ID Data |
|---|---|---|---|
| | Hex | **FE 4E** | ID Data |
| | ASCII | **■ N** | ID Data |

Creates a custom character.  Each character is divided into 8 rows of 5 pixels, each data byte represents one row.  Each byte is padded by three zero bits followed by five bits representing each pixel state.  A one represents an on condition while a zero is off.  Characters are lost when a new memory bank is loaded, unless they are saved.

| ID | 1 byte, character ID 0-7 |
|---|---|
| Data | 8 bytes, character pixel data as shown below |

*Table 14: Custom Degree Character*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data1 | 000 | p1 | p2 | p3 | p4 | p5 | 00001000 | 8 |
| Data2 | 000 | p1 | p2 | p3 | p4 | p5 | 00010100 | 20 |
| Data3 | 000 | p1 | p2 | p3 | p4 | p5 | 00001000 | 8 |
| Data4 | 000 | p1 | p2 | p3 | p4 | p5 | 00000011 | 3 |
| Data5 | 000 | p1 | p2 | p3 | p4 | p5 | 00000100 | 4 |
| Data6 | 000 | p1 | p2 | p3 | p4 | p5 | 00000100 | 4 |
| Data7 | 000 | p1 | p2 | p3 | p4 | p5 | 00000011 | 3 |
| Data8 | 000 | p1 | p2 | p3 | p4 | p5 | 00000000 | 0 |

| 1.17. Saving Custom Characters | Dec | **254 193** | Bank ID Data |
|---|---|---|---|
| | Hex | **FE C1** | Bank ID Data |

Provides access to all memory banks to create and save custom characters, graph bars, and large digits.  Any new characters saved will overwrite the old, so care should be taken when writing to any bar or digit memory bank.  Bank structure is shown below.

| Bank | 1 byte, memory bank ID (0-4) |
|---|---|
| ID | 1 byte, character ID (0-7) |
| Data | 8 bytes, character pixel data as above |

*Table 15: Custom Character Banks*

| 0 | Start-up Characters | 1 | Horizontal Bars | 2 | Vertical Bars | 3 | Medium Digits | 4 | Large Digits |
|---|---|---|---|---|---|---|---|---|---|

| 1.18. Loading Custom Characters | Dec | **254 192** | Bank |
|---|---|---|---|
| | Hex | **FE C0** | Bank |

Loads a bank of custom characters into memory for use.  Must be issued before using a bank of characters.  Alternatively, an appropriate initialize command can be used.

| Bank | 1 byte, memory bank ID (0-4) |
|---|---|

| 1.19. Save Start Up Screen Custom Characters | Dec | **254 194** | ID Data |
| --- | --- | --- | --- |
| | Hex | **FE C2** | ID Data |

Saves a custom character to memory for the start up screen or repeated use. Start up characters are displayed by sending their ID to the screen.

| ID | 1 byte, character ID (0-7) |
| --- | --- |
| Data | 8 bytes, character pixel data, see custom character example |

| 1.20. Initialize Medium Number* | Dec | **254 109** |
| --- | --- | --- |
| | Hex | **FE 6D** |
| | ASCII | ■ m |

Loads the medium number custom character bank into memory. Medium numbers must be initialized before they are used.

| 1.21. Place Medium Numbers* | Dec | **254 111** | **Row Column Digit** |
| --- | --- | --- | --- |
| | Hex | **FE 6F** | **Row Column Digit** |
| | ASCII | ■ o | **Row Column Digit** |

Places a single medium decimal digit of 2 row height and 1 column width on the display at the position specified. Medium numbers must be initialized before being placed.

| Row | 1 byte, value between 1 and 20 |
| --- | --- |
| Column | 1 byte, value between 1 and 4 |
| Digit | 1 byte, single decimal digit to display |

| 1.22. Initialize Large Numbers** | Dec | **254 110** |
| --- | --- | --- |
| | Hex | **FE 6E** |
| | ASCII | ■ n |

Loads the large number custom character bank into memory. Large numbers must be initialized before they are used.

| 1.23. Place Large Number** | Dec | **254 35** | **Column Digit** |
| --- | --- | --- | --- |
| | Hex | **FE 23** | **Column Digit** |
| | ASCII | ■ # | **Column Digit** |

Places a single large decimal digit of 4 row height and 3 column width on the display at the position specified. Medium numbers must be initialized before being placed.

| Column | 1 byte, value between 1 and 4 |
| --- | --- |
| Digit | 1 byte, single decimal digit to display |

**\*Note:** Display must be at least two character rows in height to employ this command
**\*\*Note:** Display must be at least four character rows in height to employ this command

| 1.24. Initialize Horizontal Bar | Dec | **254 104** | |
|---|---|---|---|
| | Hex | **FE 68** | |
| | ASCII | **■ h** | |

Loads the horizontal bar graph custom character bank into memory.  Horizontal bar characters must be initialized before a graph is displayed.

| 1.25. Place Horizontal Bar Graph | Dec | **254 124** | **Column Row Direction Length** |
|---|---|---|---|
| | Hex | **FE 7C** | **Column Row Direction Length** |

Places a horizontal bar graph on the screen beginning at the column and row specified.  The bar extends either right or left to the length indicated.  New bars will overwrite old.

| Column | 1 byte, value between 1 and 4 |
|---|---|
| Row | 1 byte, value between 1 and 20 |
| Direction | 1 byte, 0 for right and 1 for left |
| Length | 1 byte, length in pixels of the graph (0-100) |

| 1.26. Initialize Vertical Bar | Dec | **254 118** | |
|---|---|---|---|
| | Hex | **FE 76** | |
| | ASCII | **■ v** | |

Loads the horizontal bar graph custom character bank into memory.  Horizontal bar characters must be initialized before a graph is displayed.

| 1.27. Place Vertical Bar | Dec | **254 61** | **Column Length** |
|---|---|---|---|
| | Hex | **FE 3D** | **Column Length** |
| | ASCII | **■ =** | **Column Length** |

Places a vertical bar graph on the screen beginning at the column and row specified.  The bar extends upwards to the length indicated.  A new bar will over write the old.

| Column | 1 byte, value between 1 and 4 |
|---|---|
| Length | 1 byte, length in pixels of the graph between 0 and 32 |

## General Purpose Output

| 1.28. General Purpose Output Off | Dec | **254 86** | Number |
| --- | --- | --- | --- |
| | Hex | **FE 56** | Number |
| | ASCII | ■ **V** | Number |
| Turns the specified GPO off by supplying a zero volt output. | | | |
| Number | 1 byte, GPO (1-6) to be turned off | | |

| 1.29. General Purpose Output On | Dec | **254 87** | Number |
| --- | --- | --- | --- |
| | Hex | **FE 57** | Number |
| | ASCII | ■ **W** | Number |
| Turns the specified GPO on by supplying a five volt output. | | | |
| Number | 1 byte, GPO (1-6) to be turned off | | |

| 1.30. Set Start Up GPO State | Dec | **254 195** | **Number State** |
| --- | --- | --- | --- |
| | Hex | **FE C3** | **Number State** |
| Sets the start up state of the specified GPO.  Changes will only be seen on start up. | | | |
| Number | 1 byte, GPO (1-6) to be turned off | | |
| State | 1 byte, 1 for on or 0 for off | | |

## Display Functions

| 1.31. Display On | Dec | **254 66** | Minutes |
|---|---|---|---|
| | Hex | **FE 42** | Minutes |
| | ASCII | **■ B** | Minutes |

Turns the display backlight on for a specified length of time.  If an inverse display color is used this command will essentially turn on the text.

Minutes     1 byte, number of minutes to leave backlight on (0 for infinite)

| 1.32. Display Off | Dec | **254 70** | |
|---|---|---|---|
| | Hex | **FE 46** | |
| | ASCII | **■ F** | |

Turns the display backlight off.  If an inverse display colour is used this command will turn off the text.

| 1.33. Set Brightness* | Dec | **254 153** | Brightness |
|---|---|---|---|
| | Hex | **FE 99** | Brightness |

Immediately sets the backlight brightness.  If an inverse display color is used this represents the text colour intensity instead.  Default is 255.

Brightness     1 byte, brightness level from 0(Dim)-255(Bright)

| 1.34. Set and Save Brightness* | Dec | **254 152** | Brightness |
|---|---|---|---|
| | Hex | **FE 98** | Brightness |

Immediately sets and saves the backlight brightness.  Although brightness can be changed using the set command, it is reset to the saved value on start up.  Default is 255.

Brightness     1 byte, brightness level from 0(Dim)-255(Bright)

| 1.35. Set Contrast** | Dec | **254 80** | Contrast |
|---|---|---|---|
| | Hex | **FE 50** | Contrast |
| | ASCII | **■ P** | Contrast |

Immediately sets the contrast between background and text.  If an inverse display color is used this also represents the text brightness.  Default is 128.

Contrast     1 byte, contrast level from 0(Light)-255(Dark)

| 1.36. Set and Save Contrast** | Dec | **254 145** | Contrast |
|---|---|---|---|
| | Hex | **FE 91** | Contrast |

Immediately sets and saves the contrast between background and text.   Although contrast can be changed using the set command, it is reset to saved value on start up.  Default is 128.

Contrast     1 byte, contrast level from 0(Light)-255(Dark)

| 1.37. Set VFD Brightness*** | Dec | **254 89** | **Brightness** |
| | Hex | **FE 59** | **Brightness** |
| | ASCII | **■ Y** | **Brightness** |

Immediately sets the text brightness.  Default is 0.

| Brightness | 1 byte, brightness level from 3(Dim)-0(Bright) |

| 1.38. Set and Save VFD Brightness*** | Dec | **254 145** | **Brightness** |
| | Hex | **FE 91** | **Brightness** |

Immediately sets and saves the text brightness.  Although brightness can be changed using the set command, it is reset to the saved value on start up.  Default is 0.

| Brightness | 1 byte, brightness level from 3(Dim)-0(Bright) |

**\*Note:** OLED and LCD Model only command
**\*\*Note:** LCD Model only command
**\*\*\*Note:** VFD Model only command

## Data Security

| 1.39. Set Remember | Dec | 254 147 | Switch |
|---|---|---|---|
| | Hex | FE 93 | Switch |

Allows changes to specific settings to be saved to the display memory.  Writing to memory can be slow and each change consumes 1 write of approximately 100,000 available.  Set and save functions are always remembered.  Remember is off by default.

| Switch | 1 byte, 1 for on or 0 for off |
|---|---|

| 1.40. Set Data Lock | Dec | 254 202 245 160 | Level |
|---|---|---|---|
| | Hex | FE CA F5 A0 | Level |

Temporarily locks certain aspects of the display to ensure no inadvertent changes are made.  A new level overrides the old, and levels can be combined.  Default is 0.

| Level | 1 byte, each bit representing a level, see table below |
|---|---|

*Table 16: Data Lock Bits*

| Unlock | Reserved | Reserved | Reserved | Setting | Reserved | Command | Display |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

*Table 17: Lock Parameters*

| Unlock | Unlocks all setting locks |
|---|---|
| Reserved | Place holders only, should be 0 |
| Setting | Locks settings such as backlight and contrast |
| Command | Locks all commands |
| Display | Locks entire display, no new information can be displayed |

| 1.41. Set and Save Data Lock | Dec | 254 203 245 160 | Level |
|---|---|---|---|
| | Hex | FE CB F5 A0 | Level |

Permanently locks certain aspects of the display to ensure no inadvertent changes are made.  A new level overrides the old, and levels can be combined.  Default is 0.

| Level | 1 byte, see data lock table |
|---|---|

## Miscellaneous

| 1.42. Read Version Number | Dec | **254 54** |
|---|---|---|
| | Hex | **FE 36** |
| | ASCII | ■ 6 |

| Causes display to respond with its firmware version number. |
|---|
| Response | 1 byte, convert to hexadecimal to view major and minor revision numbers |

| 1.43. Read Module Type | Dec | **254 55** |
|---|---|---|
| | Hex | **FE 37** |
| | ASCII | ■ 7 |

| Causes display to respond with its module number. |
|---|
| Response | 1 byte, module number, see partial list below |

*Table 18: Sample Module Type Responses*

| 4F | MOS Displays | 50 | MOI Displays | 53 | MOU Displays |
|---|---|---|---|---|---|

# 9 Appendix

The versatility of the *X*-Board allows it to be interfaced to a variety of different displays.  As a result, Character Sets, Environmental tolerances, Electrical specifications, and Optical characteristics will be display dependent.  Applicable *X*-Board values are shown below for reference.

## 9.1 Environmental

*Table 19: Environmental Specifications*

| | |
|---|---|
| **Operating Temperature** | 0°C to +50°C |
| **Storage Temperature** | -20°C to +70°C |
| **Operating Relative Humidity** | Maximum 90% non-condensing |

## 9.2 Electrical

*Table 20: Electrical Specifications*

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Current Draw (Board Only) | - | 25 | - | mA |
| Maximum Single GPO Current | - | - | 20 | mA |
| Input Voltage (Vcc) | 4.75 | 5.00 | 5.25 | V |

## 9.3 Serial Interface

*Table 21: Serial Tolerances*

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Input Low Voltage | -0.5 | 0 | 0.2Vcc | V |
| Input High Voltage | 0.6Vcc | 5 | Vcc +0.5 | V |

## 9.4 I$^2$C Interface

*Table 22: I$^2$C Tolerances*

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Input Low Voltage | -0.5 | 0 | 0.3Vcc | V |
| Input High Voltage | 0.7Vcc | 5 | Vcc +0.5 | V |
| Clock Frequency | 0 | - | 400 | KHz |

## 9.5 Dimensional Drawings

Please see display datasheets for screen dependent dimensions and drawings.
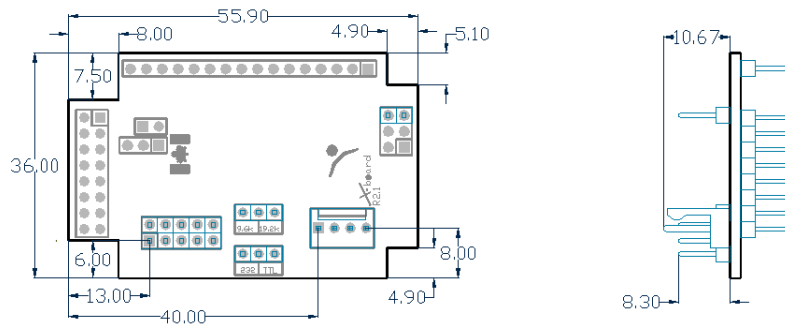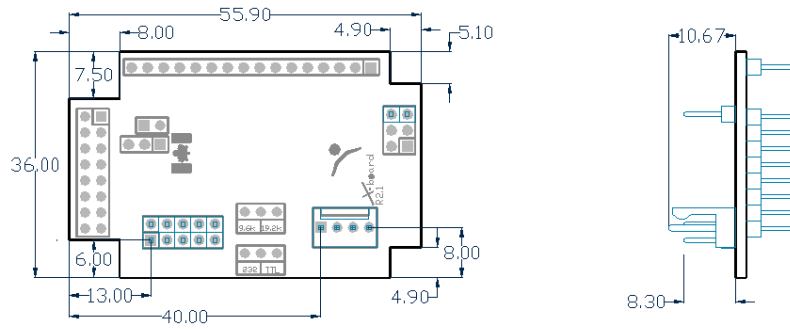


*Figure 17: MOS Model Dimensional Drawing*
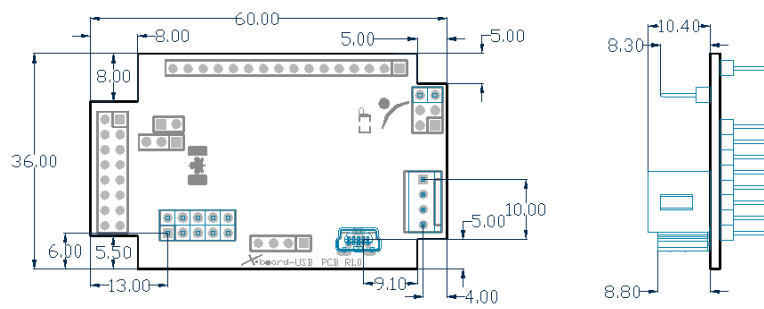


*Figure 18: MOS Model Dimensional Drawing*



*Figure 19: MOU Model Dimensional Drawing*

# 10 Ordering

## 10.1 Part Numbering Scheme

*Table 23: Part Numbering Scheme*

| MOS | -AL | -162 | F | -YX |
|-----|-----|------|---|-----|
| 1   | 2   | 3    | 4 | 5   |

## 10.2 Options

*Table 24: Display Options*

| # | Designator | Options |
|---|-----------|---------|
| 1 | Interface Type | MOI: I$^2$C protocol only<br>MOS: Serial protocol using RS232 or TTL levels<br>MOU: USB protocol only |
| 2 | Display Type | AL: Alphanumeric Liquid Crystal Display<br>AO: Alphanumeric Organic Light Emitting Diode<br>AV: Alphanumeric Vacuum Florescent Display |
| 3 | Display Size | 162: Sixteen columns by two rows<br>164: Sixteen columns by four rows<br>202: Twenty columns by two rows<br>204: Twenty columns by four rows<br>242: Twenty-four columns by two rows |
| 4 | Display Model | A: Standard Size (162 and 204 Models Only)<br>B: Larger Screen and Text (204 Models Only)<br>C: Standard Size (202 Models Only)<br>F: Smaller Screen and Text (162 Model Only) |
| 5 | Colour | YX: Yellow-Green Background with Grey Text<br>BW: Blue (Inverse) Background with White Text<br>WB: White Background with Blue-Grey Text<br>XR: Black (Inverse) Background with Red Text<br>XY: Black (Inverse) Background with Yellow-Green Text<br>FA: Black (FFSTN Inverse) Background with Amber Text<br>FB: Black (FFSTN Inverse) Background with Blue Text<br>FG: Black (FFSTN Inverse) Background with Green Text<br>FW: Black (FFSTN Inverse) Background with White Text<br>OB: OLED Blue on black<br>OG: OLED Green on black<br>OR: OLED Red on black<br>OW: OLED White on black<br>OY: OLED Yellow on black |

Please note that the X-Board can be mated to any parallel Matrix Orbital alphanumeric display.  Check the Parallel Display section at www.matrixorbital.com for a complete list of displays.
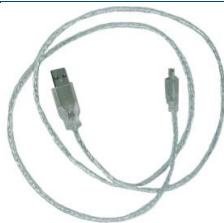
## 10.3 Accessories

**Power**

*Table 25: Power Accessories*

| PCS | Standard Power Cable |  |
|---|---|---|

**Communication**

*Table 26: Communication Accessories*

| EXTMUSB3FT | Mini-USB Cable |  |
|---|---|---|
| INTMUSB3FT | Internal Mini-USB Cable |  |
| SCCPC5V | Serial Communication/5V Power Cable |  |
| BBC | Breadboard Cable |  |

## Mounting

| | | |
|---|---|---|
| **B2021-XX**** | 20x2 Mounting Bracket with Coloured Overlay |  |
| **B202A-XX*** | 20x2 Mounting Bracket with Keyed Overlay |  |
| **B2041-XX**** | 20x4 Mounting Bracket with Coloured Overlay |  |
| **B204A-XX*** | 20x4 Mounting Bracket with Keyed Overlay |  |
| **MK2021-XX**** | 20x2 Coloured Overlay with Drill Guide |  |
| **MK2041-XX**** | 20x4 Coloured Overlay with Drill Guide |  |

**Note:** No mounting options are available for 16x2 displays.

**\*Note:**  Keyed overlays are available in Black Vinyl (-BK) and Beige Vinyl (no part extension).  Deep Red (-DR) and Silver (-SK) options are available for 20x2 sized displays only.

**\*\*Note:**  Non-keyed overlays are available in Black Vinyl (-BK), Black Brushed Aluminum (-BL), Silver Brushed Aluminum (-AL), and Beige Vinyl (no part extension).

# 11 Definitions

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

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

FFSTN: Double film super-twisted nematic in reference to an LCD.  The addition of two layers of film between the STN display and polarizer improves contrast.

GPO:   General purpose output, used to control peripheral devices from a display.

GUI:   Graphical user interface.

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

$I^2$C:   Inter-integrated circuit protocol employing a clock and data line to communicate a short distance at slow speeds between a master and up to 128 addressable slave devices.  A Matrix Orbital display is a slave device.

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

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

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

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

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

STN:   Super-twisted nematic in reference to an LCD.  In a relaxed or nematic state, crystals orientate themselves in the same direction and allow light to pass.  In an excited state these crystals align to block light.  Super-twisted crystals move from 180 to 270 degrees between phases to providing contrast than TN models.

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

USB:   Universal Serial Bus protocol widely used in PCs.

# 12 Contact

**Sales**
Phone: 403.229.2737
Email: sales@matrixorbital.ca

**Support**
Phone: 403.204.3750
Email: support@matrixorbital.ca

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