



Technical Manual

Revision1.1

PCB Revision: 1.0 or greater

Firmware Revision: 1.0 or greater

Revision History

Revision	Description	Author
1.0	Initial Release	Clark
1.1	Added VFD Option	Clark

Table of Contents

Introduction....2
Quick Start....3
- Parts Included....3
- Basic Connections....4
- USB....5
- Serial....6
- I2C....7

Software....8
- Numeric Conversion....8
- Hyperterminal....8
- uProject....9
- Code (C#)....9

Hardware....10
- Advanced Connections....10
- Screen....11
- Serial DB9 Connector....11
- USB Connectors....12
- Power/Communication Header....12
- AC Power Input....13
- Protocol Select....13
- Keypad....14
- General Purpose Output....14
- Dallas One-Wire....15
- Manual Override....15

Troubleshooting....16
- Power....16
- Display....16
- Communication....16
- Manual Override....16

Commands....17
Appendix....34
- Character Set....34
- Environmental....35
- Electrical....35
- Optical....35
- Dimensional Drawing....36

Ordering....37
- Options....37
- Accessories....38

Production Products....41
- Display Options....41

Definitions....43
Contact Info....44



Introduction

The Matrix Orbital DevDevil is a product development board which provides all of the functionality of an alphanumeric display in a project development friendly package. A variety of color options are offered for both the 2 row by 20 column and 4 row by 20 column screen models.

All DevDevil products contain the same 6 general purpose outputs and 5 x 5 keypad input found on their production counterparts. In addition to the regular 5V, 20mA outputs, indicator LEDs can be enabled beside each GPO to provide status indication. Alternate keys are also added in a tactile style layout, sharing the same pins as the regular keypad. The keypad header also has the same power and ground pins to provide optional grounding or backlight support.

The DevDevil also offers RS232, USB, TTL, and I2C protocols through a dynamic switch block, rather than the more rigid solder jumpers found on the production model. Maximum communication speeds, however, remain the same for these protocols; 153800bps for Serial protocols and 100,00Kbps for I2C. Standard headers for all protocols are offered on the board, as well as a variety of input power options.

Powering up the DevDevil can be as easy as connecting to the mini-USB connector on board. Alternate power options include an AC adapter, power through the DB9 header, and 4 pin data/power input. The basic USB connection can be used to easily interface to the development board while the additional options are useful for more advanced project requirements.

The following pages will outline these features in greater detail and provide instruction for connecting and communicating to the Matrix Orbital DevDevil. Using just the information provided and a little imagination, the DevDevil can open the door to a world of exciting display possibilities!

Quick Start

Parts Included

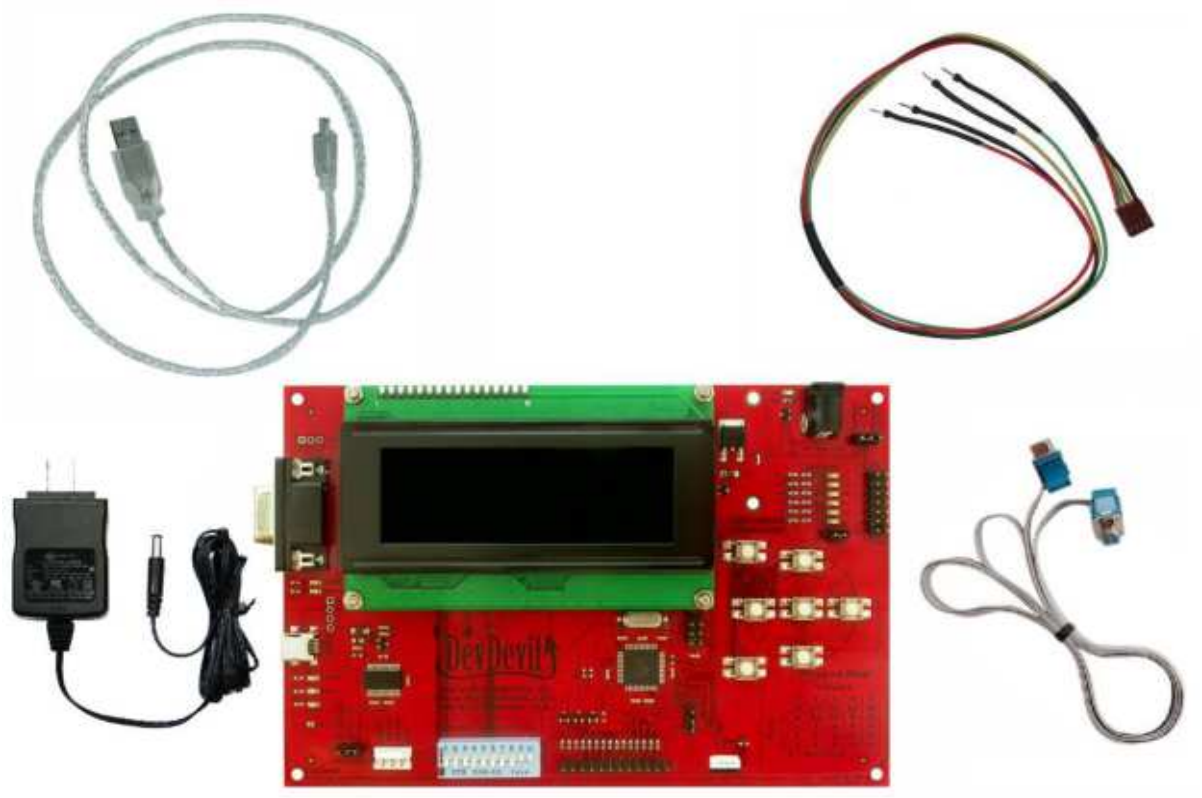


Illustration 1: Included and Optional Parts*

1 x	DevDevil Development Board	DD-20X-XX
1 x	3ft mini-B USB Cable	EXTMUSB3FT
1 x	Breadboard Development Cable**	BBC
1 x	AC Power Adapter**	ACPC
1 x	4ft Serial Cable**	CSS4FT

Table 1: Included and Recommend Parts List*

***Note:** If any parts are missing from your order, please contact a sales associate as quickly as possible using the Contact Information at the end of this document.

****Note:** Recommended optional equipment available for purchase to use in specific applications, see below

Basic Connections

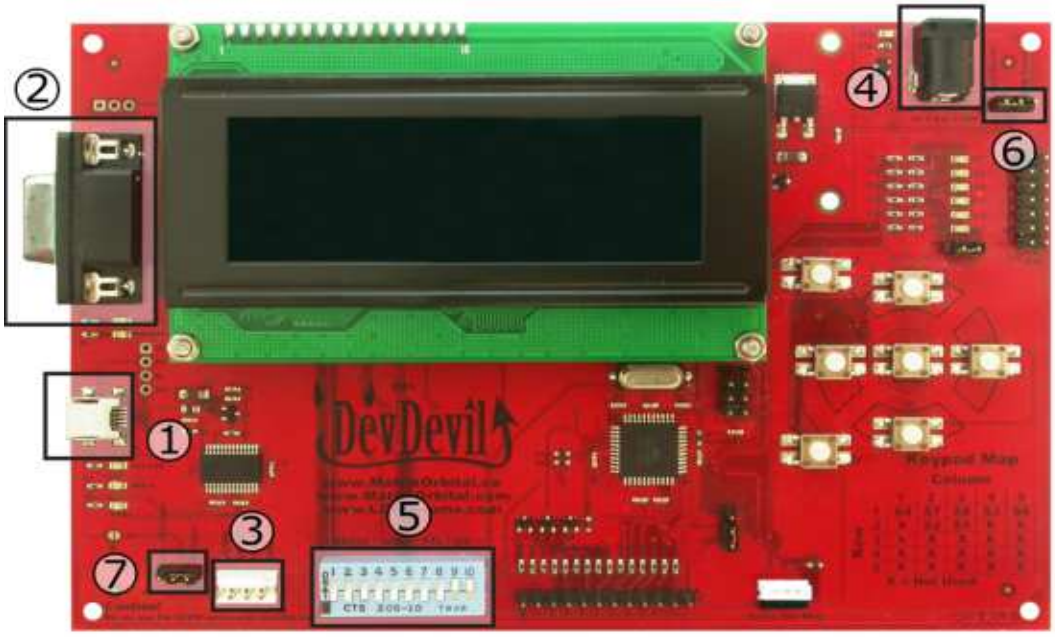


Illustration 2: DevDevil Basic Connections

1	USB Connector	5	Protocol Select Switches
2	Serial DB9 Connector	6	Input Select Jumper
3	Power/Communication Header	7	Voltage Select Jumper
4	AC Power Input		

Table 2: DevDevil Basic Connections

USB

Cables Required	EXTMUSB3FT



Illustration 3: USB Connection

1	2	3	4	5	6	7	8	9	10
			X	X	X				
X	X	X				X	X	X	X

Table 3: USB Protocol Select

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

1. Prepare the onboard jumpers.
   i. Ensure that Input Select jumper is set to USB, see highlight 6 in Illustration 2: DevDevil Basic Connections..
   ii. Disconnect the Voltage Select jumper, see highlight 7 in Illustration 2: DevDevil Basic Connections..
2. Set the Protocol Select switches.
   i. Place the unit into USB protocol by moving switches 4, 5, and 6 up into the on position, while ensuring the remainder are down in the off position, see highlight 5 in Illustration 2: DevDevil Basic Connections..
3. Make the connections.
   i. Plug the mini-B header of your external mini USB cable into the DevDevil and the regular USB header into your computer USB jack, see highlight 1 in Illustration 2: DevDevil Basic Connections..
4. Install the drivers.
   i. Download the latest drivers at www.matrixorbital.ca/drivers, and save them to a known location.
   ii. When prompted, install the USB bus controller driver automatically.
   iii. If asked, continue anyway, even though the driver is not signed.
   iv. When the driver install is complete, your display will turn on, but communication will not yet be possible.
   v. At the second driver prompt, install the serial port driver automatically.
   vi. Again, if asked, continue anyway.
5. Create.
   i. Use uProject or hyperterminal to get started, then move on with your own development. Instructions for the former and a simple C# example is available for the latter can be located in the Software section.

Serial

Cables Required	ACPC, CSS4FT

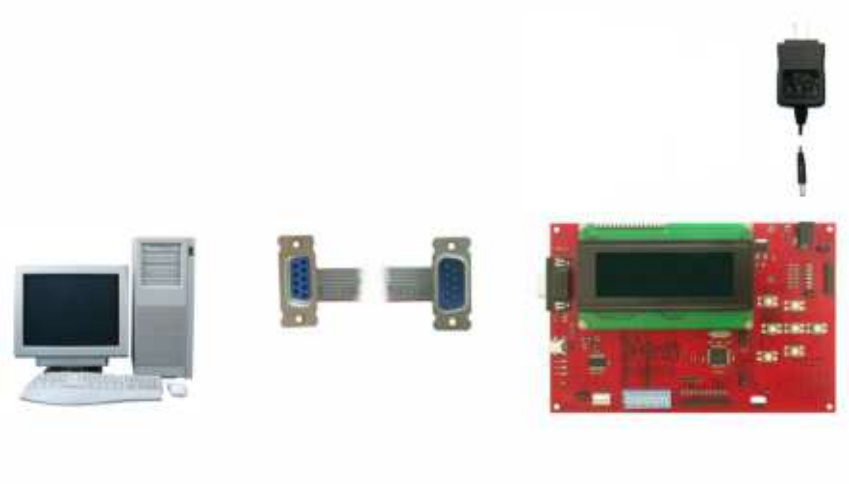


Illustration 4: RS232 Connection

1	2	3	4	5	6	7	8	9	10
X	X	X							
			X	X	X	X	X	X	X

Table 4: RS232 Protocol Select

Serial protocol provides a classic alternative the the USB method listed above. Two additional cables are required for this connection; the 4 ft serial cable and AC power adapter. These must be purchased in addition to the DevDevil kit. This set up will provide constant regulated power to your development board and provide a familiar communication scheme. To place your board in Serial mode, just adhere to the steps laid out below.

1. Prepare the onboard jumpers.
    i. Select Input Power on the Input Select jumper, see highlight 6 in Illustration 2: DevDevil Basic Connections..
    ii. Remove the Voltage Select jumper, see highlight 7 in Illustration 2: DevDevil Basic Connections..
2. Set the Protocol Select switches.
    i. Place the unit into Serial protocol by moving switches 1, 2, and 3 up into the on position, while ensuring the remainder are down in the off position, see highlight 5 in Illustration 2: DevDevil Basic Connections..
3. Make the connections.
    i. Insert the male end of your serial cable into the corresponding header on the unit and mate the female connector with the desired communication port of your computer, see highlight 3 in Illustration 2: DevDevil Basic Connections..
    ii. Place the output power plug of your AC power adapter into the onboard barrel jack and connect the business end of the device into an available wall socket, see highlight 4 in Illustration 2: DevDevil Basic Connections..
4. Create.
    i. Again, uProject or hyperterminal will get started, then move on with your own development. Instructions for the former and a simple C# example is available for the latter can be located in the Software section.

I²C

Cables Required	BBC

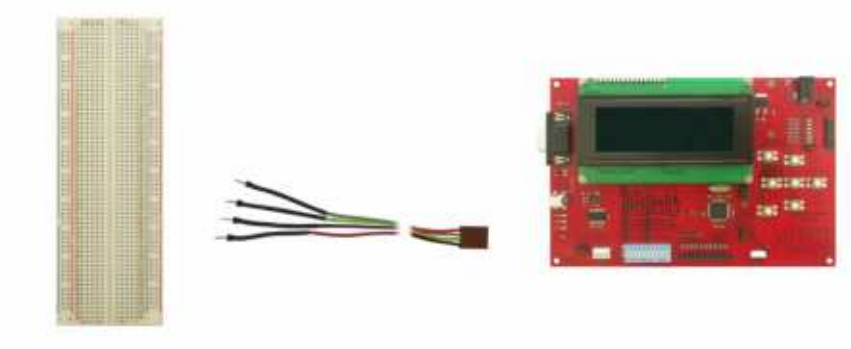


Illustration 5: I²C Connection

1	2	3	4	5	6	7	8	9	10
								X	X
X	X	X	X	X	X	X	X		

Table 5: I²C Protocol Select

The most advanced connection to the DevDevil is provided by the I²C protocol setting. This mode is usually used when connecting to a controller and serial TTL capability is not available. It requires a connection directly to your project, best accomplished using a breadboard and the included cable of the same name. Power is usually applied from the breadboard, but can also be supplied directly to the DevDevil using the optional AC power adapter. To dive right into your application and use the DeveDevil in I²C mode, get started with the guidelines below.

1. Prepare the onboard jumpers.
   i. Connect the Input Power side of the Input Select jumper, see highlight 6 in Illustration 2: DevDevil Basic Connections..
   ii. Select the appropriate Voltage Select jumper placement according to input voltage supplied, see highlight 7 in Illustration 2: DevDevil Basic Connections..
2. Set the Protocol Select switches.
   i. Place the unit into I²C protocol by moving switches 9 and 10 up into the on position, while ensuring the remainder are down in the off position, see highlight 5 in Illustration 2: DevDevil Basic Connections..
3. Make the connections.
   i. Connect the breadboard cable header to the power/communication header on your DevDevil and plug the four leads into your breadboard. The red lead will require power, while the black should be connected to ground. The green and yellow should be connected to your controller clock and data lines respectively, see highlight 2 in Illustration 2: DevDevil Basic Connections..
   ii. If required, power can be drawn from a wall source by removing the Voltage Select jumper. Place the output power plug of your AC power adapter into the onboard barrel jack and connect the business end of the device into an available wall socket, see highlight 4 in Illustration 2: DevDevil Basic Connections..
4. Create.
   i. 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's a breeze to switch over to another protocol, and fellow developers are always on our forums, http://www.lcdforums.com/forums, for additional support.

Software

Numeric Conversion

All command values listed in the document are shown in decimal notation for continuity and simplicity, however, different applications my require different numeric systems. The most basic computing numeric system is binary notation. Each digit is a placeholder for a value equaling 2 to the power of it's position. Decimal notation can be converted using the table below by placing a one in the largest placeholder value, subtracting that placeholder number from the decimal number, then using the remaining number to repeat the process. The decimal number 88 is written as follows.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	**1**	**0**	**1**	**1**	**0**	**0**	**0**

Table 6: Binary Conversion

0	1	0	1	1	0	0	0
128	64	32	16	8	4	2	1
5				**8**			

Table 7: Hexadecimal Conversion

From binary, numbers can be easily converted to hexadecimal for quick interpretation and manipulation. Four bits are converted to a single character from 0 to F, meaning a byte can be written with just two characters. These can then be used in a look up table to locate corresponding ASCII characters, which represent bytes with a single digit. The website www.ascii-table.com is a good reference for numeric conversions, including all of those mentioned above.

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, select the appropriate communication port that your display is connected to. Finally, the port settings below must be entered to complete the port setup.

BPS	Data Bits	Parity	Stop Bits	Flow Control
19200	8	None	1	None

Table 8: Hyperterminal Settings

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. Select properties from the file menu and open 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**

Table 9: Hyperterminal Command Format

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

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 **/**

Table 10: uProject Raw Data Format

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 more 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 preform correctly when a USB display is connected. Please manually configure any USB display.

Code (C#)

Full demonstration programs and code are available for the DevDevil 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.

These programs are meant to showcase the capability of the DevDevil 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.

Hardware

Advanced Connections

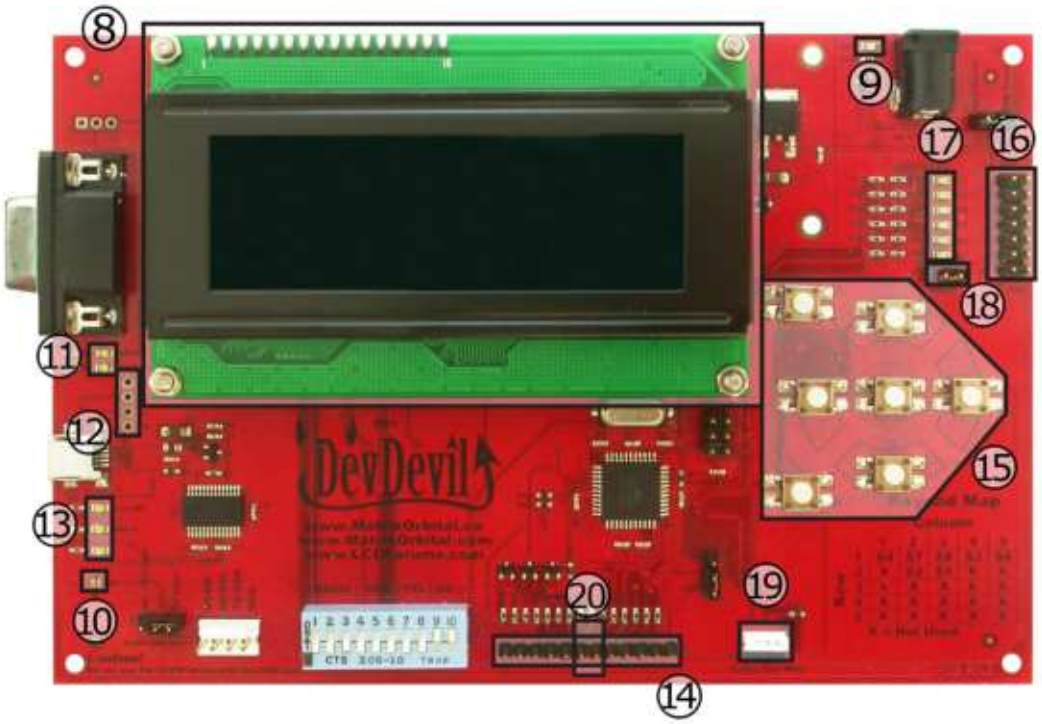


Illustration 6: DevDevil Advanced Connections

8	Screen*	15	Keypad Buttons
9	Power Indicator	16	GPO Header
10	Optional Power Through DB9 Jumper**	17	GPO Indicators
11	Serial Communication Indicators	18	GPO Indicator Enable
12	Optional Alternate USB Header**	19	Dallas One-Wire Connector
13	USB Communication Indicators	20	Manual Override
14	Keypad Header		

Table 11: DevDevil Advanced Connections

***Note:** The 4x20 LCD screen shown, the DevDevil is also available with a 2x20 sized display

****Note:** Non-standard headers that can be populated by custom request only

Screen

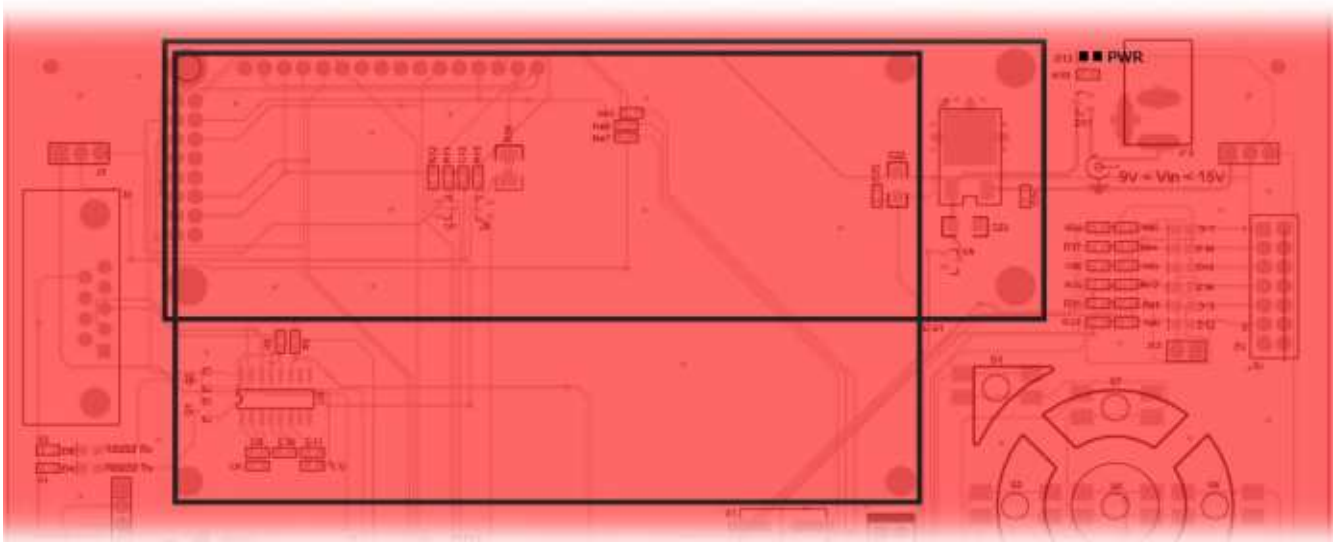


Illustration 7: Screen Location and Power Indicator

The DevDevil provides a twenty character column alphanumeric display that is either two or four rows high. Both of these popular screens are available in an array of colours on a variety of Production modules listed in the appendix below. The initial power up message, contrast, and brightness settings are customizable.

Serial DB9 Connector

Mating Cables	CSS1FT, CSS4FT

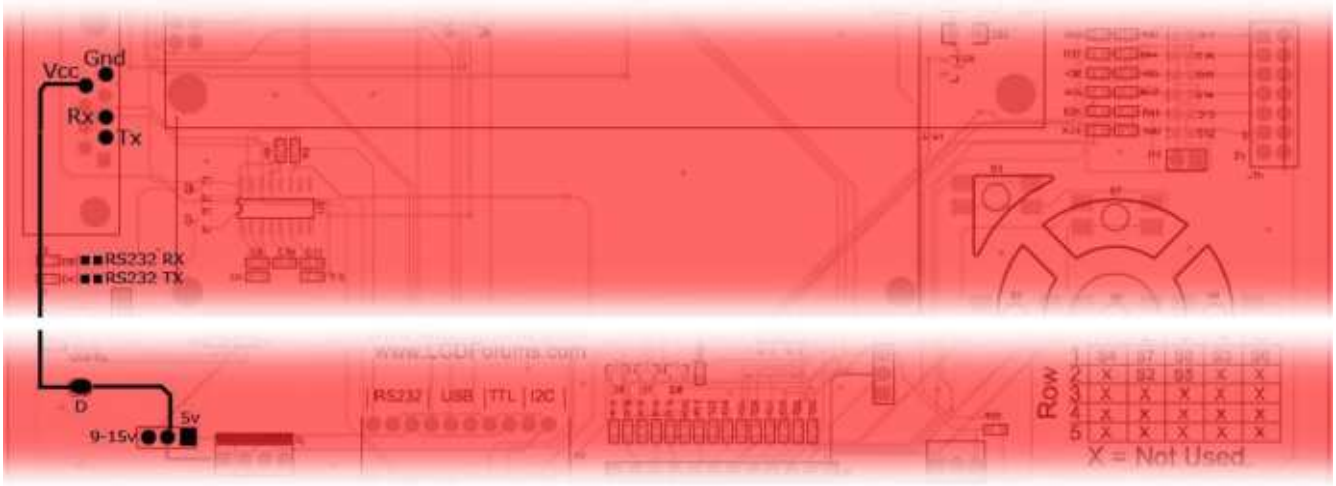


Illustration 8: Serial DB9 Connector, Power Option, and Indicators

The serial protocol is most easily accessed using DB9 serial header onboard. This connection provides receive, transmit, and ground pins with an option to supply power to the entire board as well. Using a custom cable with power supplied to pin 9, either 5V or 9-15V can be supplied to the board once the 'D' jump is soldered. It is also important to set the appropriate pins on the Voltage Select jumper and ensure the Power Select jumper is set to input.

USB Connectors

Mating Cables	EXTMUSB3FT, INTMUSB3FT

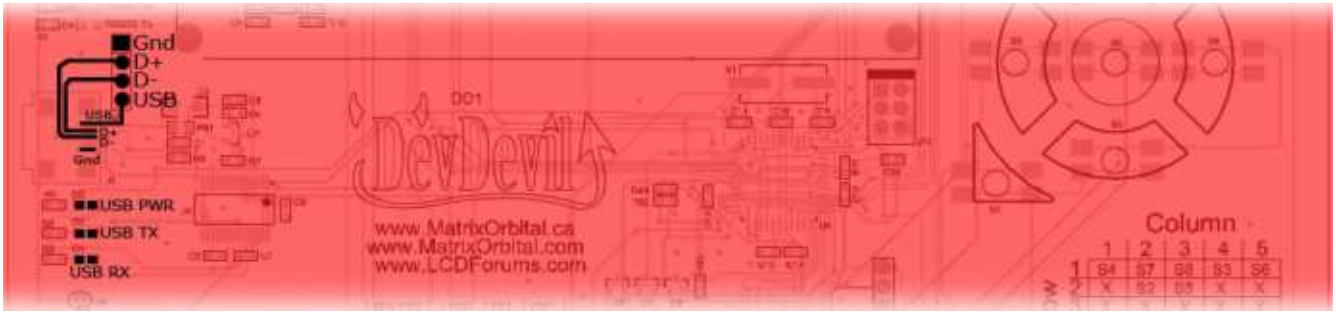


Illustration 9: Mini and Alternate USB Connections with Indicators

The mini USB configuration is supplied for communication and power input. To order the DevDevil with the alternate USB header populated for more advanced applications, contact a Matrix Orbital sales representative. In order to power the DevDevil using USB, ensure that the Voltage Select jumper is removed and the Power Select jumper is set to USB. Please see the USB quick start guide for additional information.

Power/Communication Header

Mating Cables	BBC, SCCPC5V, PCUSB, PC5V, PC12V



Illustration 10: Power/Communication Header and Voltage Select

The Power and Communication Header is the most versatile connector on the DevDevil. As mentioned in the quick start guide, It can provide a serial or I^2C communication connection and offers power and ground inputs for power. Input power to this connector can be either 5V or 9-15V, however, it is important that the Voltage Select jumper is set to match. The Power Select jumper should also be set to input. Please see the serial quick start guide for more information.

AC Power Input

Mating Cables	ACPC



Illustration 11: AC Power Jack and Input Select

In addition to the methods mentioned above, the DevDevil can be powered with an AC to DC adapter. The output voltage of any device must be between 9 and 15 volts DC and the barrel used must be center positive. For this configuration, the Voltage Select jumper must be placed on the 9-15V Input side and the Power Select jumper set to input. This input can be useful when communicating to the device via a computer serial port.

Protocol Select



Illustration 12: Protocol Select Switches

To change the communication protocol used on the DevDevil, easy to use switches are provided. Three switches are required for both RS232 and USB protocols, while only two are needed for TTL or I2C communication. To make a selection, push all the switches in the desired block up, toward the centre of the board while pulling all others down.

Keypad

Mating Cables	KPP4X4

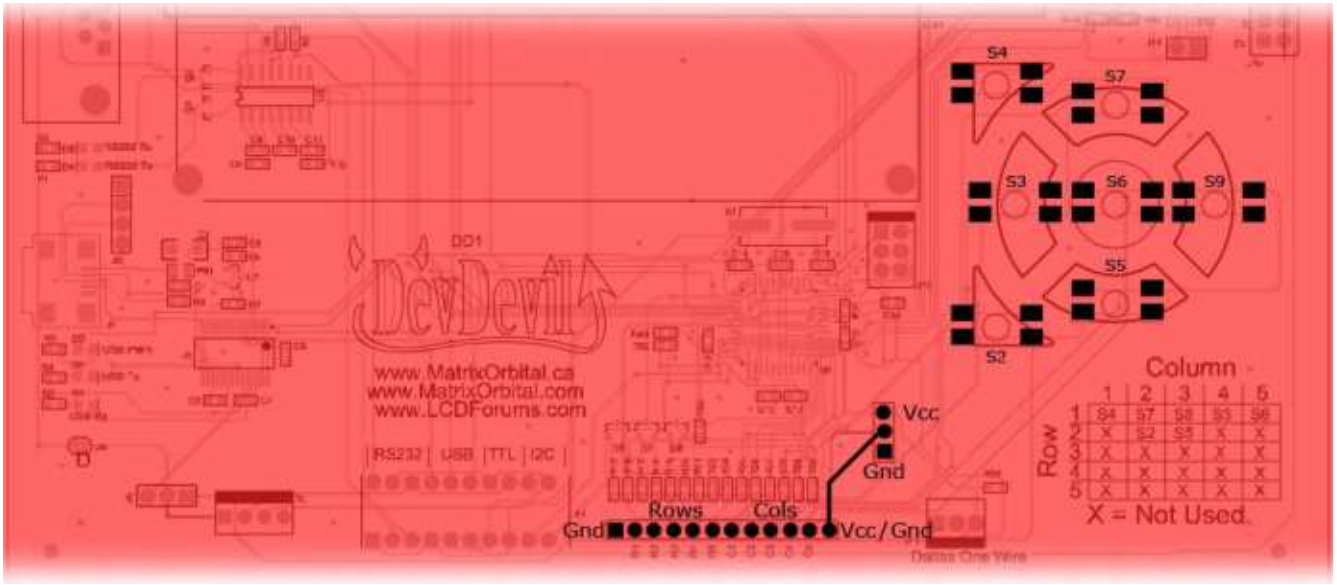


Illustration 13: Keypad Header and Buttons

Facilitating user input, the keypad interface header allows a Matrix Orbital or any other passive matrix style input device to be attached to the DevDevil in a 5x5 row, column format. A customizable value is returned to the host each time one of the five row pins is connected to a column pin. Two additional pins are offered on either side of these inputs to provide a single ground and a selectable ground or power pin to provide grounding or backlight capabilities to an external input device. Alternatively, some of the row and column inputs are doubled on the keypad layout onboard for quick testing as per the legend provided on the module.

General Purpose Output



Illustration 14: General Purpose Outputs, Indicators, and Indicator Enable

A total of six devices can be driven using the general purpose outputs available on the DevDevil. Each output 5V, 20mA output can be software toggled. Indicator LEDs can be activated by jumping LED Enable. These outputs are most commonly drive transistor circuitry to provide switching capability.

Dallas One-Wire

Mating Cables	Temperature Probe



Illustration 15: Dallas One-Wire Connector

The Dallas One-Wire header affords the ability to add a variety of devices to the DevDevil for purposes ranging from temperature sensing to external memory. Multiple devices can be connected to this bus and accessed using the DevDevil. Only non-parasitic devices, those drawing power from the dedicated pin rather than the data line, have been tested, however, the data line has been pulled high through the nearby resistor to facilitate parasitic operation. A simple temperature monitoring application has been provided using the Matrix Orbital probe.

Manual Override

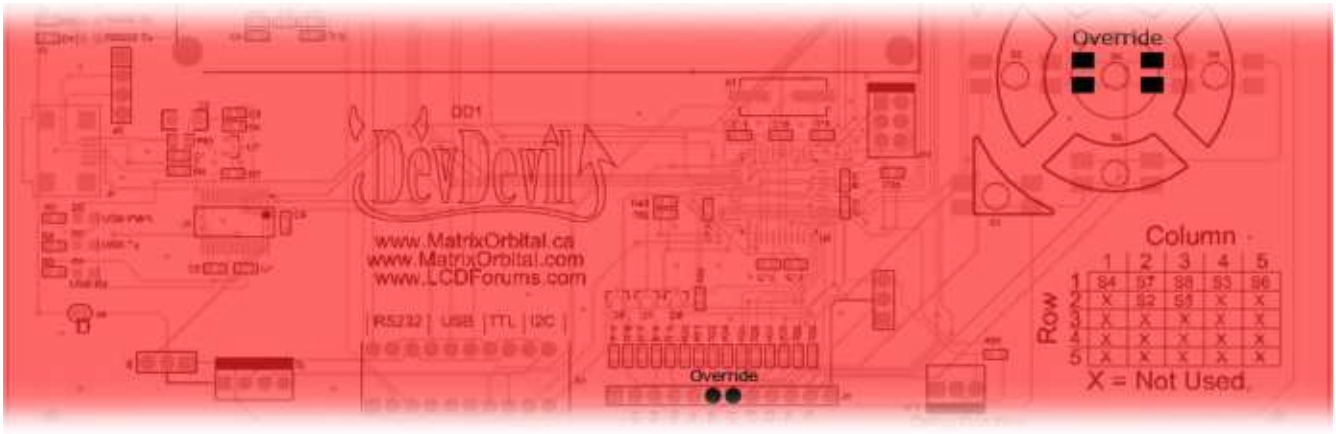


Illustration 16: Manual Override

A Manual Override feature is available on the DevDevil to override all settings to their factory defaults. This override can be extremely useful if the device becomes unresponsive or behaves in an undesirable fashion. To override the unit to the values below, see the Manual Override entry in the Troubleshooting section below. Please note that the settings will be temporary and will revert when power is cycled once more, so it is recommended that desired settings are saved.

Baud Rate	19200
I²C Address	0x50
Backlight	255
Contrast	128

Table 12: Manual Override Settings

Troubleshooting

Power

In order for the DevDevil to function correctly, it must be supplied with the appropriate power. If the D10 power LED near the AC power adapter input is not illuminated, power is not applied correctly. Try following the tips below.

- When using the mini USB header for power, ensure that no jumper is present on the J6 Input Voltage selection.
- The AC power adapter input also requires no jumper on the Input Voltage Select jumper.
- If power is applied through the DB9 connector, ensure 'D' is connected and the appropriate 5V or 9-15V Input Voltage Selection is jumped at J6.
- Power applied through the 4 pin power/communication header must also be specified using the appropriate 5V or 9-15V Voltage selection.
- Ensure all the appropriate protocol select switches are pressed up into the on position and all unused protocol switches are down.
- Select the appropriate power source by selecting USB power at J14 for a USB connection or Input power for all other sources.

Display

If the DevDevil 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 manual override below.
- 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.

Communication

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

- Ensure the baud rate of the DevDevil and host match.
- Unlock the display. See the Set and Save Data Lock command for more info.
- Match Rx from the DevDevil to the transmitting pin from your host and the Tx pin to the receiving pin.

Manual Override

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

1. Disconnect power from your DevDevil.
2. Place a jumper on the middle two pins of the keypad header, R5 and C1.
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 Table 12: Manual Override Settings. 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.

Commands

1. Communications

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

Immediately changes the I^2C write address.

Address 1 byte, even value

1.2.Changing the Baud Rate			
	Dec	**254 57**	Speed
	Hex	**FE 39**	Speed
	ASCII	**■ 9**	Speed

Immediately changes the baud rate. Can be temporarily forced to 19200 with manual override.

Speed 1 byte, valid settings shown below

Rate	1200	2400	4800	9600	19200	28800	38400	57600
Speed	83	41	207	103	51	34	25	16

1.3.Setting a Non-Standard Baud Rate			
	Dec	**254 164**	Speed
	Hex	**FE A4**	Speed

Immediately changes the baud rate to a non-standard value. Range is 977 to 153800. Speed must be a whole number. Due to rounding, error increases with baud rate, actual baud must be within 3% of desired baud to ensure accurate communication. Baud rate can be temporarily forced to 19200 by a manual override.

Speed 2 byte, calculations shown below

$$Speed = \frac{CrystalSpeed}{(8 \times DesiredBaud)} - 1 \qquad ActualBaud = \frac{CrystalSpeed}{(8 \times (Speed + 1))}$$

$$\frac{|DesiredBaud - ActualBaud|}{DesiredBaud} < 0.03$$

1.4.Transmission Protocol Select			
	Dec	**254 160**	Protocol
	Hex	**FE A0**	Protocol

Selects the protocol used for data transmission to the host. Data transmission from the host is not affected. Must be set to the protocol in use to receive data correctly.

Protocol 1 byte, 1 for Serial (RS232/TTL/USB) or 0 for I^2C

2. Text

2.1.Control Characters*							
8 (08Hex)	Backspace	10 (0AHex)	Line feed/New line	12 (0CHex)	Clear screen/New page	13 (0DHex)	Carriage return

***Note:** Regular text can be sent using regular alphanumeric strings or individual character values, starting at 65.

2.2.Auto Scroll On		
	Dec	**254 81**
	Hex	**FE 51**
	ASCII	**■ Q**
Entire contents of screen are shifted up one line when the end of the screen is reached. Display default is auto scroll on.		

2.3.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.		

2.4.Clear Screen		
	Dec	**254 88**
	Hex	**FE 58**
	ASCII	**■ X**
Clears the contents of the screen.		

2.5.Changing the Startup Screen			
	Dec	**254 64**	**Characters**
	Hex	**FE 40**	**Characters**
	ASCII	**■ @**	**Characters**
Changes the message displayed on startup. Custom characters can be included by adding their decimal value (0-7). Characters will automatically wrap on the display.			
Characters	40 bytes for 20x2 screen, 80 bytes for 20x4		

2.6.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. Display default is auto line wrap on.

2.7.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. 20X4 screens will write to rows 1, 3, 2, then 4 while 20x2 screens will write to rows 1 then 2 but hide one row of characters after each line. Display default is auto line wrap on.

2.8.Set Cursor Position	Dec 254 71 Column Row Hex FE 47 Column Row ASCII ■ G Column Row

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

Column	1 byte, value between 1 and 20
Row	1 byte, value between 1 and 2 for 20x2 screen or 1 and 4 for 20x4 screen

2.9.Go Home	Dec 254 72 Hex FE 48 ASCII ■ H

Returns the cursor to the top left of the screen.

2.10.Move Cursor Back	Dec 254 76 Hex FE 4C ASCII ■ L

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

2.11.Move Cursor Forward	Dec 254 77 Hex FE 4D ASCII ■ M

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

2.12.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.

2.13.Underline Cursor Off	Dec	254 75
	Hex	FE 4B
	ASCII	■ K

Removes line under current cursor position.

2.14.Blinking Block Cursor On	Dec	254 83
	Hex	FE 53
	ASCII	■ S

Displays a blinking block over the current cursor position. Can be used with underline.

2.15.Blinking Block Cursor Off	Dec	254 84
	Hex	FE 54
	ASCII	■ T

Removes blinking block over current cursor position.

3. Special Characters

3.1.Create 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

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

3.2.Save 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

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

3.3.Load 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)

3.4.Save Startup 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

3.5.Initialize Medium Numbers	Dec	254 109
	Hex	FE 6D
	ASCII	■ m

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

3.6.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 2 for 20x2 screen or 1 and 4 for 20x4 screen
Digit	1 byte, single decimal digit to display

3.7.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 use.

3.8.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

3.9.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.

3.10.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 2 for 20x2 screen or 1 and 4 for 20x4 screen
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)

***Note:** This command is available for 20x4 screens only

3.11.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.

3.12.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 2 for 20x2 screen or 1 and 4 for 20x4 screen
Length	1 byte, length in pixels of the graph (0-16 for 20x2 and 0-32 for 20x4)

4. General Purpose Output

4.1.General Purpose Output Off			
	Dec	254 86	Number
	Hex	FE 56	Number
	ASCII	■ V	Number
Turns the specified GPO off. Will only affect onboard LEDs if LED enable is jumped.			
Number	1 byte, GPO (1-6) to be turned off		

4.2.General Purpose Output On			
	Dec	254 87	Number
	Hex	FE 57	Number
	ASCII	■ W	Number
Turns the specified GPO on. Will only affect onboard LEDs if LED enable is jumped.			
Number	1 byte, GPO (1-6) to be turned off		

4.3.Set Startup 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		

5. Dallas 1-Wire

5.1.Search for a 1-Wire Device	Dec 254 200 2 Hex FE C8 02
Sends a search query to all of the up to 32 devices on the one wire bus. Any connected device will respond with a identification packet.	
Response	14 bytes, identification packet as shown below

Offset	Length	Value	Description
0	2	9002	Preamble
2	1	138	Another device packet will follow OR
		10	Last device packet
3	1	49	Packet Type
4	1	0	Error Code (0 indicates success)
5	8		Device Address
13	1	0	CRC8 address check (0 indicates validity)

5.2.Dallas 1-Wire Transaction	Dec	254 200 1	Flags Send Bits Receive Bits Data
	Hex	FE C8 01	Flags Send Bits Receive Bits Data

Preforms a single Dallas 1-Wire transaction. Consult your device documentation for information regarding device specific protocols. If an error is encountered, a corresponding value will be returned by the device.

Flags	1 byte, flags for transaction, see below
Send Bits	1 byte, number of bytes to be sent to the device
Receive Bits	1 byte, number of bytes expected to be received from the device
Data	Variable, data to be transmitted LSB to MSB

Bit	Flag Description
7	
6	Unused
5	
4	0 (Future Compatibility)
3	Add CRC8 to transaction
2	0 (Future Compatibility)
1	Read CRC8 from transaction
0	Reset Bus prior to transaction

Code	Error Description
0	Success
1	Unknown Command
2	No Devices Found
3	Fatal Search Error

6. Keypad

6.1.Auto Transmit Key Presses On	Dec	254 65
	Hex	FE 41
	ASCII	■ A

Key presses are automatically sent to the host when received by the display. Default is auto transmit on.

6.2.Auto Transmit Key Presses Off	Dec	254 79
	Hex	FE 4F
	ASCII	■ O

Key presses are held in the 10 key buffer to be polled by the host. Use this mode for I^2C transactions. Default is on.

6.3.Poll Key Press	Dec	254 38
	Hex	FE 26
	ASCII	■ &

Reads the last unread key press from the 10 key display buffer. If another key is stored in the buffer the MSB will be 1, the MSB will be 0 when the last key press is read. If there are no stored key presses a value of 0 will be returned. Auto transmit key presses must be turned off for this command to be successful.

Response	1 byte, value of key pressed (MSB determines additional keys to be read)

6.4.Clear Key Buffer	Dec	254 69
	Hex	FE 45
	ASCII	■ E

Clears all key presses from the display buffer.

6.5.Set Debounce Time	Dec	254 85 Time
	Hex	FE 55 Time
	ASCII	■ U Time

Sets the time between a key press and a key read by the display. Most switches will bounce when pressed, the debounce time allows the switch to settle for an accurate read. Default is 8 representing a debounce time of approximately 52ms.

Time	1 byte, debounce increment (debounce time = time * 6.554ms)

6.6.Set Auto Repeat Mode	Dec	254 126	Mode
	Hex	FE 7E	Mode

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

Mode	1 byte, 1 for hold mode or 0 for typematic

6.7.Auto Repeat Mode Off	Dec	254 96
	Hex	FE 60

Turns auto repeat mode off. Default is on (typematic).

6.8.Assign Keypad Codes	Dec	254 213	Key Down Key Up
	Hex	FE D5	Key Down Key Up

Assigns the key down and key up values sent to the host when a key press is detected. Defaults are shown below and can be reset by setting all values to 255.

Key Down	25 bytes, key down values
Key Up	25 bytes, key up values

KeyDown				
A(65)	B(66)	C(67)	D(68)	E(69)
F(70)	G(71)	H(72)	I(73)	J(74)
K(75)	L(76)	M(77)	N(78)	O(79)
P(80)	Q(81)	R(82)	S(83)	T(84)
U(85)	V(86)	W(87)	X(88)	Y(89)

KeyUp				
a(97)	b(98)	c(99)	d(100)	e(101)
f(102)	g(103)	h(104)	i(105)	j(106)
k(107)	l(108)	m(109)	n(110)	o(111)
p(112)	q(113)	r(114)	s(115)	t(116)
u(117)	v(118)	w(119)	x(120)	y(121)

7. Display Functions

7.1.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)

7.2.Display Off		
	Dec	254 70
	Hex	FE 46
	ASCII	■ F

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

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

Immediately sets the VFD text brightness. Default is 0.

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

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

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

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

***Note:** **VFD** model only command

7.5.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 brightness instead. Default is 255.

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

7.6.Set and Save Brightness*	Dec	254 152	Brightness
	Hex	FE 98	Brightness

Immediately sets and saves the backlight brightness. If an inverse display colour is used this represents the text instead. Although brightness can be changed using the set command, it is reset to saved value on start up. Default is 255.

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

7.7.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)

7.8.Set and Save Contrast*	Dec	254 145	Contrast
	Hex	FE 91	Contrast

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

***Note:** LCD model only command

8. Data Security

8.1.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. Off is default.

Switch	1 byte, 1 for on or 0 for off

8.2.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

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

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

8.3.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

9. Miscellaneous

9.1.Read Version Number	Dec 254 54 Hex FE 36 ASCII ■ 6
Causes display to respond with it's firmware version number.	
Response	1 byte, convert to hexadecimal to view major and minor revision numbers

9.2.Read Module Type	Dec 254 55 Hex FE 37 ASCII ■ 7
Causes display to respond with it's module number.	
Response	1 byte, module number, see partial list below

23	DD-DB1-AL202	24	DD-DB1-AV202	25	DD-DB1-AL204	26	DD-DB1-AV204

Appendix

Character Sets

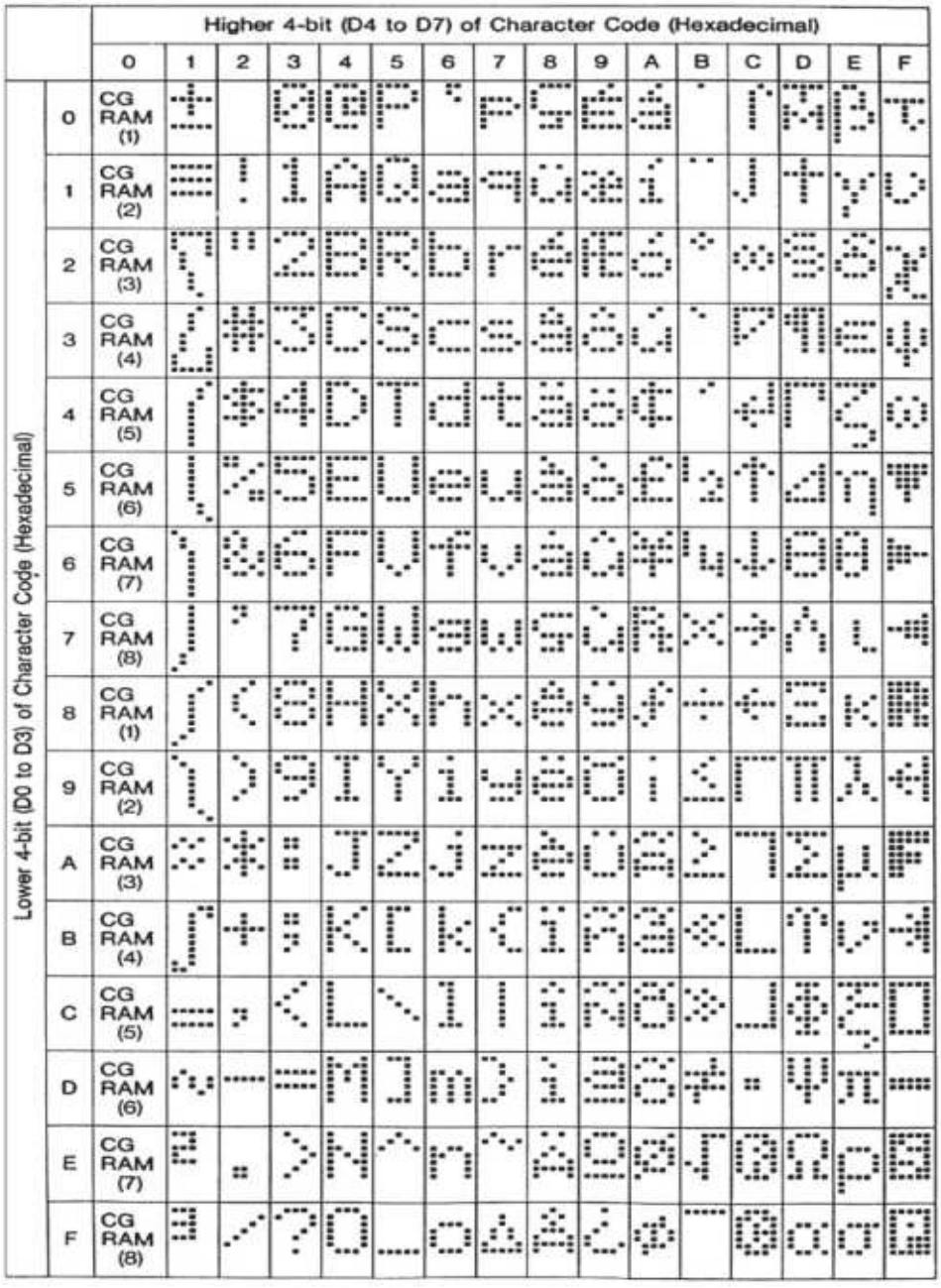


Illustration 17: DevDevil LCD Series Character Table



Illustration 18: DevDevil VFD Series Character Table

Environmental

	Standard	VFD Standard
Operating Temperature	0°C to +50°C	-20°C to +70°C
Storage Temperature	-10°C to +60°C	-40°C to +80°C
Operating Relative Humidity	Maximum 90% non-condensing	Maximum 80% non-condensing

Table 13: DevDevil Environmental Specifications

Electrical

Current Consumption

Board	+	Backlight	+	Indicator GPOs	+	Header GPOs
50mA		20-115mA		1.25mA each		20mA each maximum

Table 14: DevDevil Current Consumption

Backlight Consumption

	YG	GW	WB	IY	IR	FW	FB	VFD*
20x2	105mA	40mA	45mA	N/A	115mA	30mA	30mA	100mA
20x4	70mA	20mA	20mA	70mA	N/A	N/A	N/A	100mA

Table 15: Backlight Current Consumption by Model

Optical

	20x2	20x4	20x2 VFD	20x4 VFD	units
Screen Size	116.0 x 37.0	98.0 x 60.0	116.0 x 37.0	98.0 x 60.0	mm
Viewing Area	83.0 x 18.6	76.0 x 25.2	N/A		mm
Active Area	73.5 x 11.5	70.4 x 20.8	70.8 x 11.5	70.8 x 20.9	mm
Character Size	3.20 x 5.55	2.95 x 4.75	2.50 x 4.70	2.40 x 4.70	mm
Character Pitch	3.70 x 5.95	3.55 x 5.35	3.60 x 6.10	3.60 x 5.40	mm
Pixel Size	0.60 x 0.65	0.55 x 0.55	0.40 x 0.50	0.40 x 0.50	mm
Pixel Pitch	0.65 x 0.70	0.60 x 0.60	0.50 x 0.70	0.50 x 0.70	mm
Viewing Angle	-30 to +30		N/A		°
Luminance	N/A		350		cd/m^2
Backlight Half-Life	50,000		N/A		Hours

Table 16: DevDevil Optical Characteristics

***Note:** A VFD style display will draw up to twice the normal operating current at start up inrush.

Dimensional Drawing

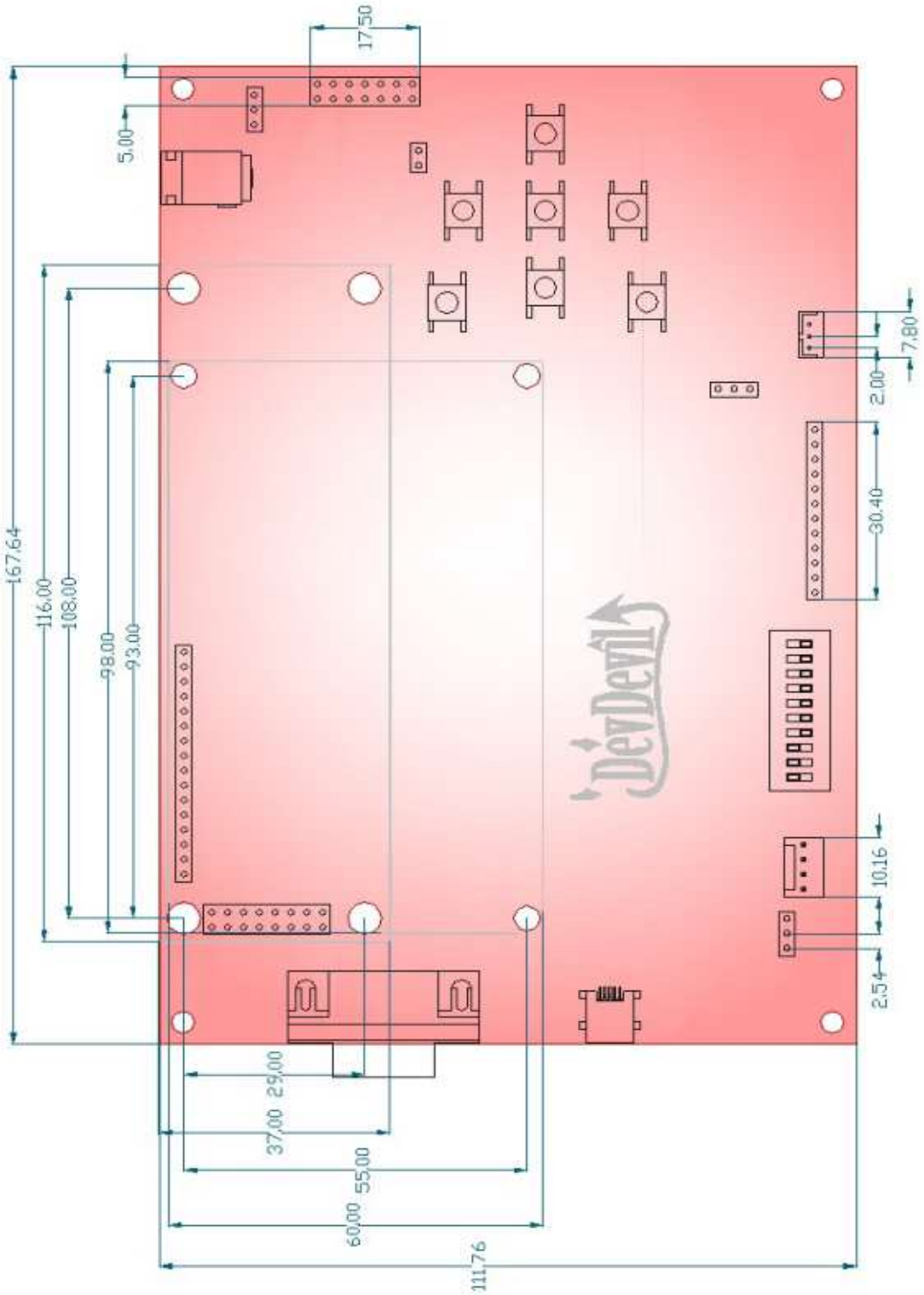


Illustration 19: DevDevil Dimensional Drawing

Ordering

Options

DD	-DB1	-AL	202	C	-GW
1	2	3	4	5	6

#	Designator	Options
1	Product Line	DevDevil
2	Product Type	Development Board, Revision 1.0
3	Display Type	-AL: Alphanumeric Liquid Crystal Display -AV: Alphanumeric Vacuum Florescent Display
4	Display Size	202: 20 rows by 2 columns 204: 20 rows by 4 columns
5	Screen Size	A: 98.0mm x 60.0mm (204 only) C: 116.0mm x 37.0mm (202 only)
6	Color	NP: Standard Yellow-green background with grey text -GW: White background with grey text -WB: Blue background with white text (inverse display) -IY**: Yellow-green background with black text (inverse display) -IR*: Red background with black text (inverse display) -FW*: Black background with white text (inverse FFSTN display) -FB*: Black background with blue text (inverse FFSTN display)

***Note:** Available on 20x2 display models only
****Note:** Available on 20x4 display models only

Accessories

Power

ACPC*	AC Power Converter	
PC5V	5V Power Cable	
PC12V	12V Power Cable	
PCUSB	USB Power Cable	

Table 17: DevDevil Power Accessories

***Note:** Recommended Optional Accessory

Communication

CSS1FT	1 ft. Serial Cable	
CSS4FT**	4 ft. Serial Cable	
EXTMUSB3FT*	Mini-USB Cable	
INTMUSB3FT	Internal Mini-USB Cable	
SCCPC5V	Serial Communication/5V Power Cable	
BBC**	Breadboard Cable	

Table 18: DevDevil Communication Accessories

***Note:** Included Accessory
****Note:** Recommended Optional Accessory

Peripherals

KPP4X4	16 Button Keypad	
Temperature Probe	Dallas One-Wire Temperature Probe	

Table 19: DevDevil Peripheral Accessories

Production Products

Display Options

Protocol*

	LK/VK202-25	LK/VK202-25-USB	LK/VK204-45	LK/VK204-25-USB
USB	Not Available	Default	Not Available	Default
RS232	Default	Not Available	Default	Not Available
TTL	Available	Not Available	Available	Not Available
I2C	Available	Not Available	Available	Not Available

Table 20: Production Protocol Options

Input Voltage

	LK/VK202-25	LK/VK202-25-USB	LK/VK204-45	LK/VK204-25-USB
5V (Regular)	Default	Default	Default	Default
9-15V (-V)	Available	Not Available	Available	Not Available
9-35V (-VPT)	Not Available	Not Available	Available	Not Available

Table 21: Production Voltage Options

Connectors

	LK/VK202-25	LK/VK202-25-USB	LK/VK204-45	LK/VK204-25-USB
DB9 Serial	Default	Not Available	Default	Not Available
Mini-USB**	Not Available	Default (Vertical)	Not Available	Default (Vertical)
Power/Data	Default	Not Available	Default	Not Available
AC Power	Not Available	Not Available	Not Available	Not Available
PC Power***	Not Available	Default	Not Available	Default

Table 22: Production Connector Options

***Note:** Protocol is selected using zero ohm jumper resistors, soldering is required change protocol. Available protocols can be selected at purchase.

****Note:** Mini-USB header is orientated vertically rather than horizontally.

*****Note:** PC or floppy cable power is not available on the DevDevil but offered on the production USB displays.

Peripherals

	LK/VK202-25	LK'VK202-25-USB	LK/VK204-45	LK/VK204-25-USB
Keypad*	1x12 Header	1x12 Header	1x12 Header	1x12 Header
GPO**	2x7 Header	2x7 Header	2x7 Header	2x7 Header
DOW***	Available	Default	Available	Default

Table 23: Production Peripheral Options

***Note:** Only the 1x12 keypad header is available on production units, a keypad such as the KPP4X4 unit must be purchased to interface with the display.

****Note:** Only the 2x7 GPO header is available on production units, indicator LEDs are not available. Additional indicators such as Rx, Tx, and power LEDS are also not offered on production models.

*****Note:** The Dallas One-Wire header is available on production USB units, but must be specially ordered for non-USB models.

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.

DOW: Dallas One-Wire protocol is similar to I^2C providing reduced data rates at a greater distance. One wire is provided for data, while two others supply power and ground. Matrix Orbital displays are only tested with non-parasitic devices, those that do not draw power from the data line, however, some parasitic devices may work.

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^2C: 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.

LCD: Liquid crystal display.

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 widely used serial protocol. A low level is -30V while a high is +30V.

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

SCL: Serial clock line used to designate data bits in I^2C protocol. This open drain line should be pulled high through a resistor. Nominal values are between 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 a serial protocol. A low level is 0V while high logic is 5V.

USB: Universal Serial Bus protocol widely used in PCs.

VFD: Vacuum florescent display.

Contact Info

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