



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





DevDevil

Technical Manual

Revision 1.1

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

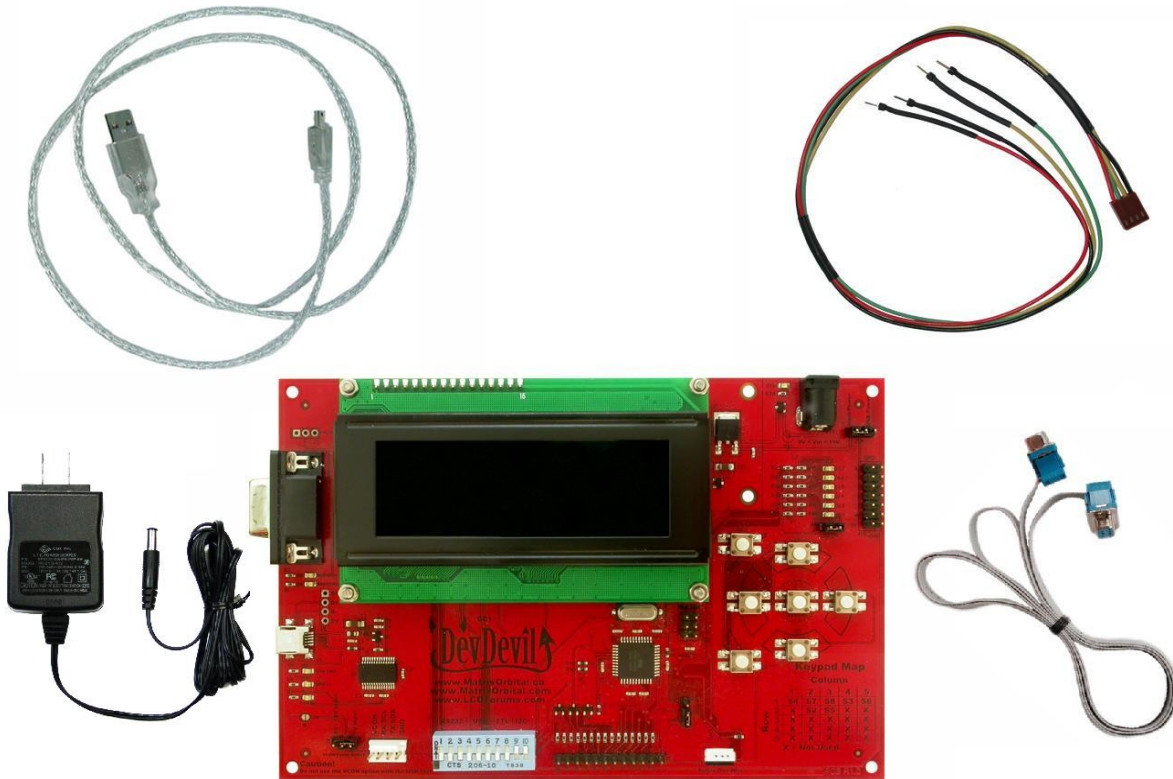


Illustration 1: Included* and Optional Parts

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

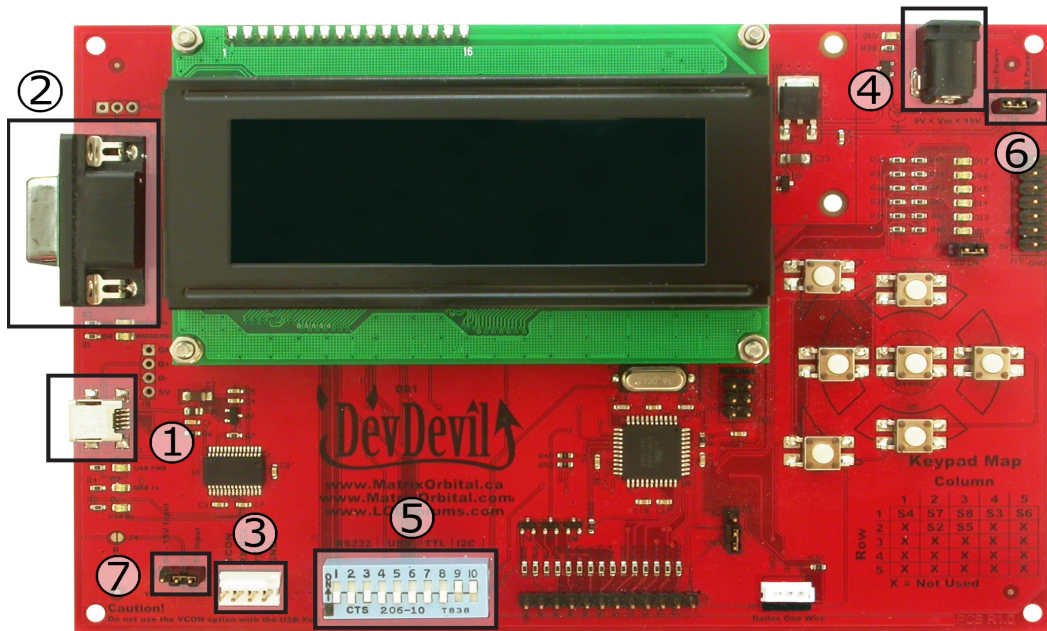


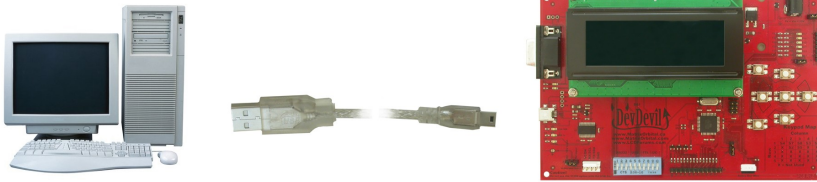
Illustration 2: DevDevil Basic Connections

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 EXTUSB3FT



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

Illustration 3: USB Connection

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

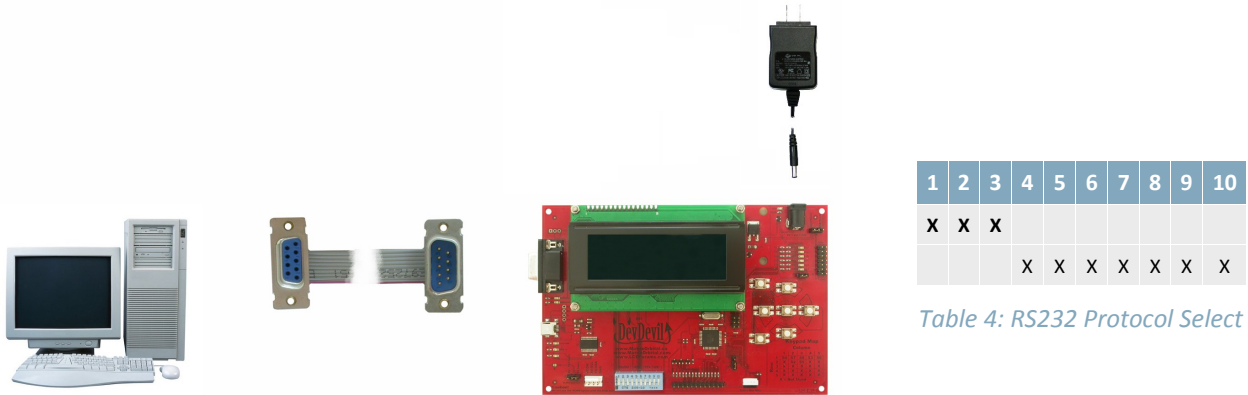


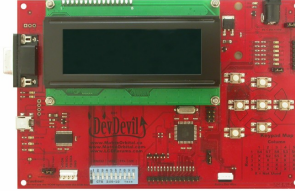
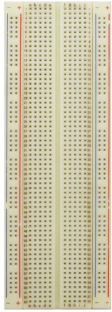
Illustration 4: RS232 Connection

Serial protocol provides a classic alternative to 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



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

Illustration 5: I²C Connection

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 DevDevil 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 may 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 its 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

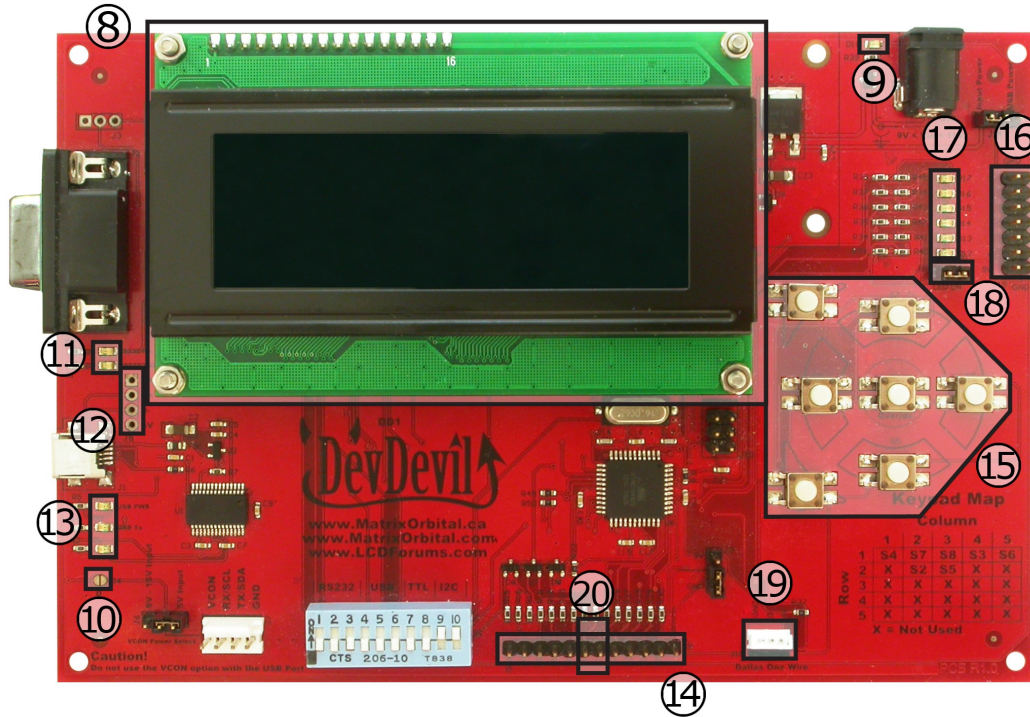


Illustration 6: DevDevil Advanced Connections

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

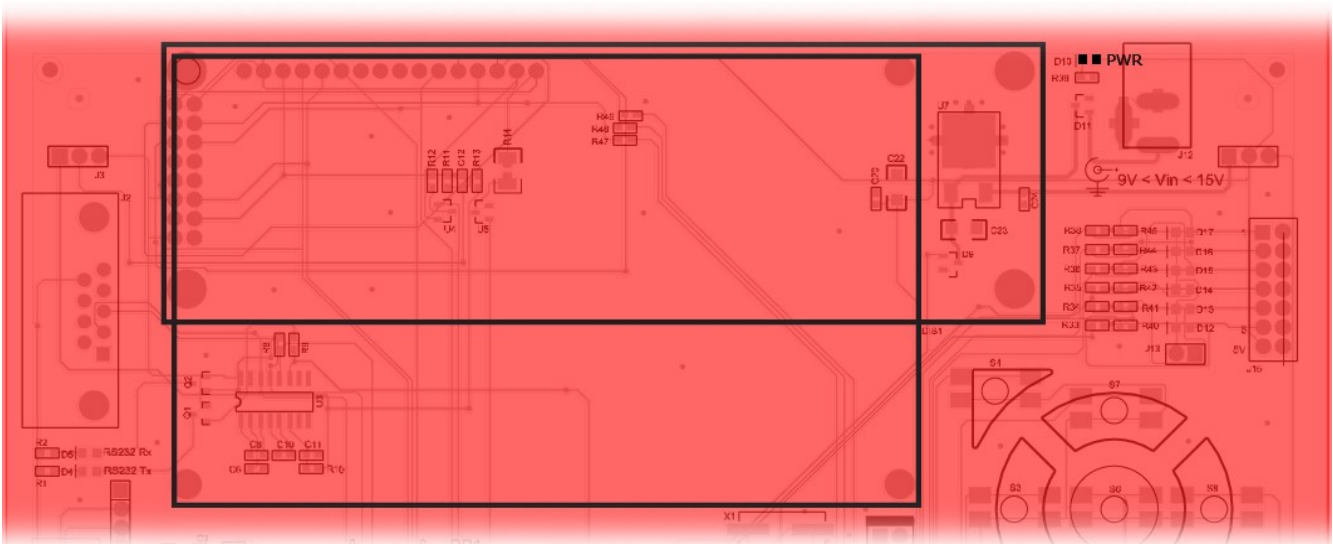


Illustration 7: Screen Location and Power Indicator

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

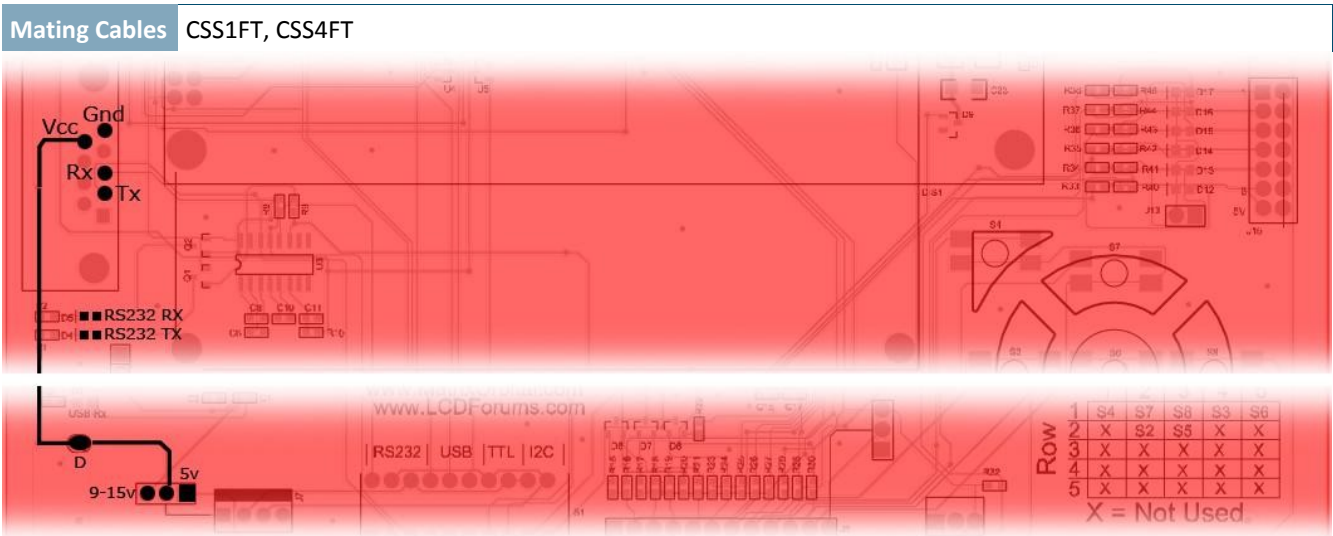


Illustration 8: Serial DB9 Connector, Power Option, and Indicators

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 EXTUSB3FT, INTMUSB3FT

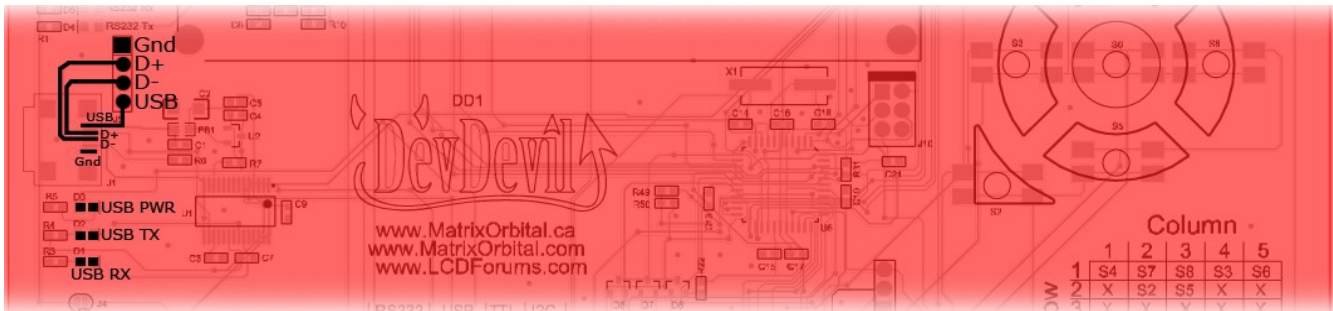


Illustration 9: Mini and Alternate USB Connections with Indicators

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

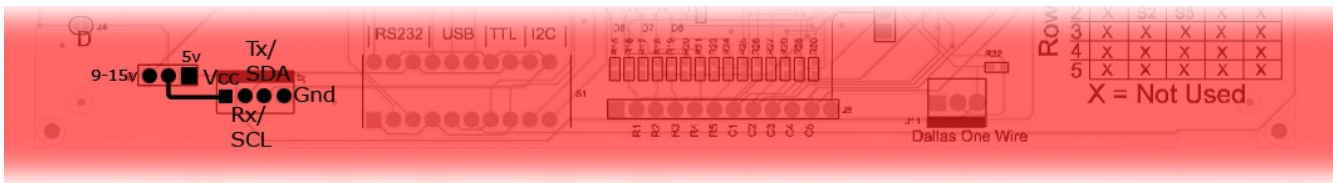


Illustration 10: Power/Communication Header and Voltage Select

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 I²C 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

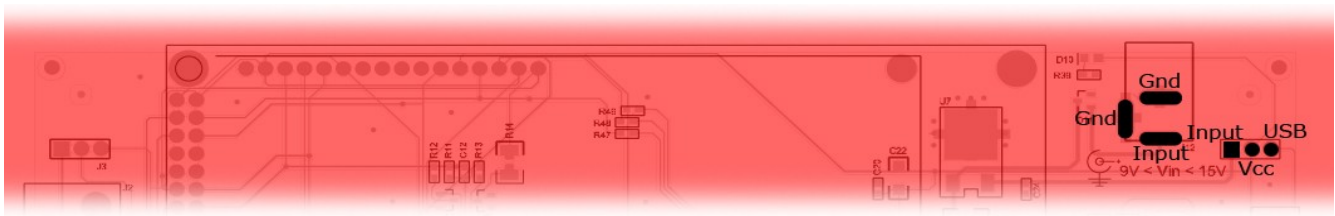


Illustration 11: AC Power Jack and Input Select

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

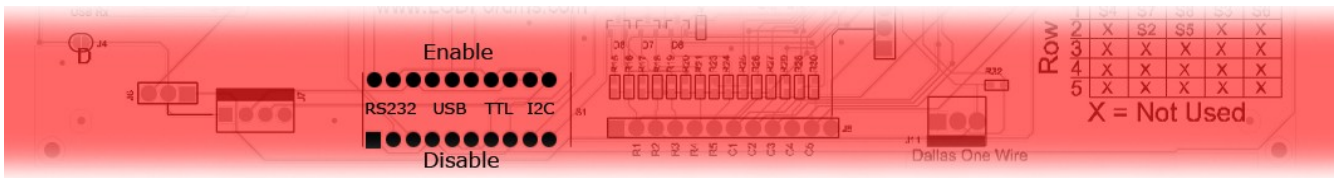


Illustration 12: Protocol Select Switches

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

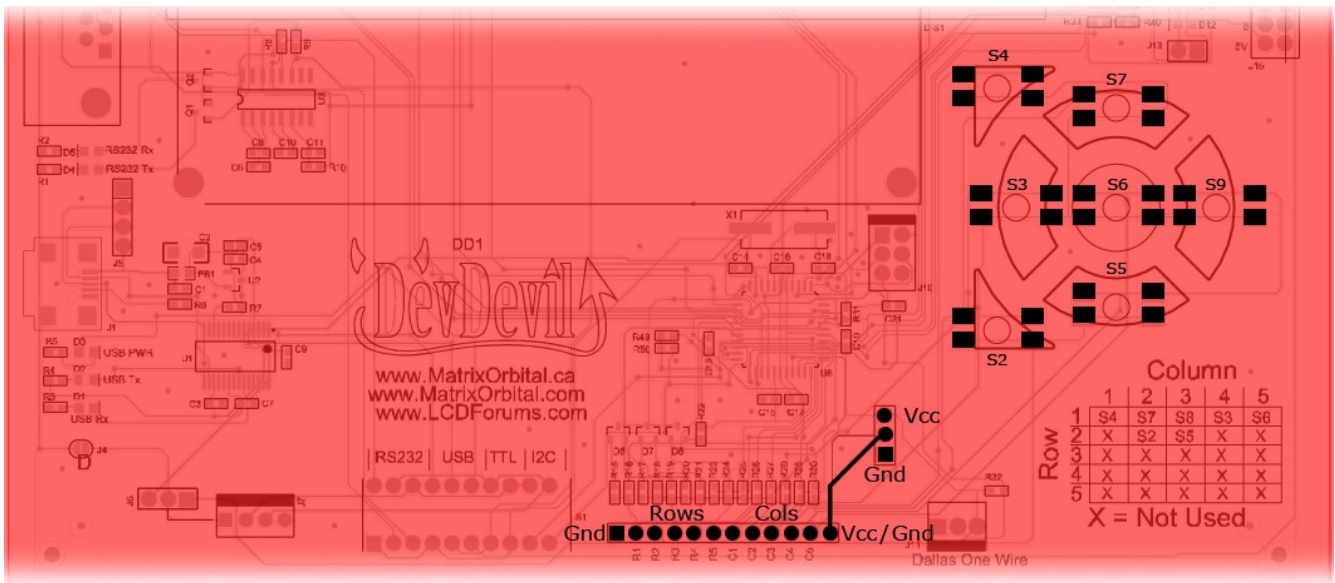


Illustration 13: Keypad Header and Buttons

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

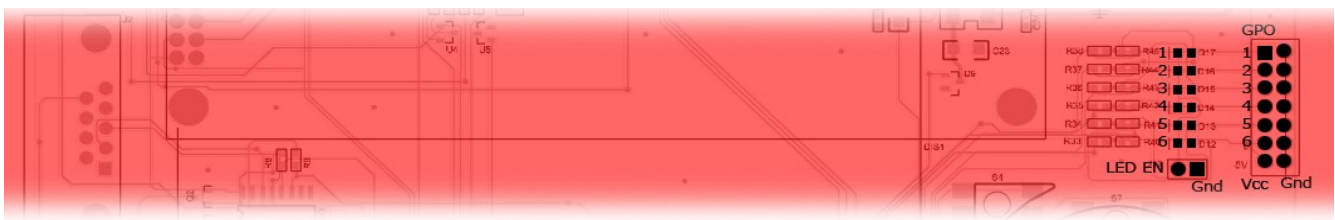


Illustration 14: General Purpose Outputs, Indicators, and Indicator Enable

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

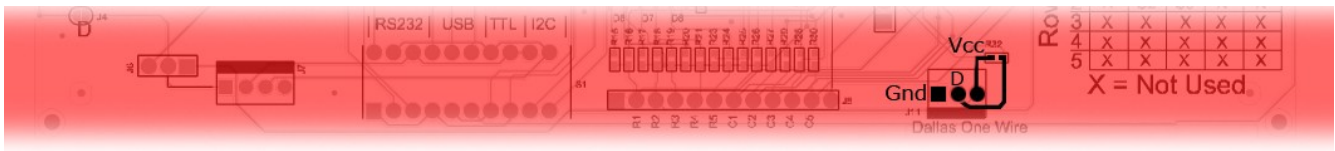


Illustration 15: Dallas One-Wire Connector

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

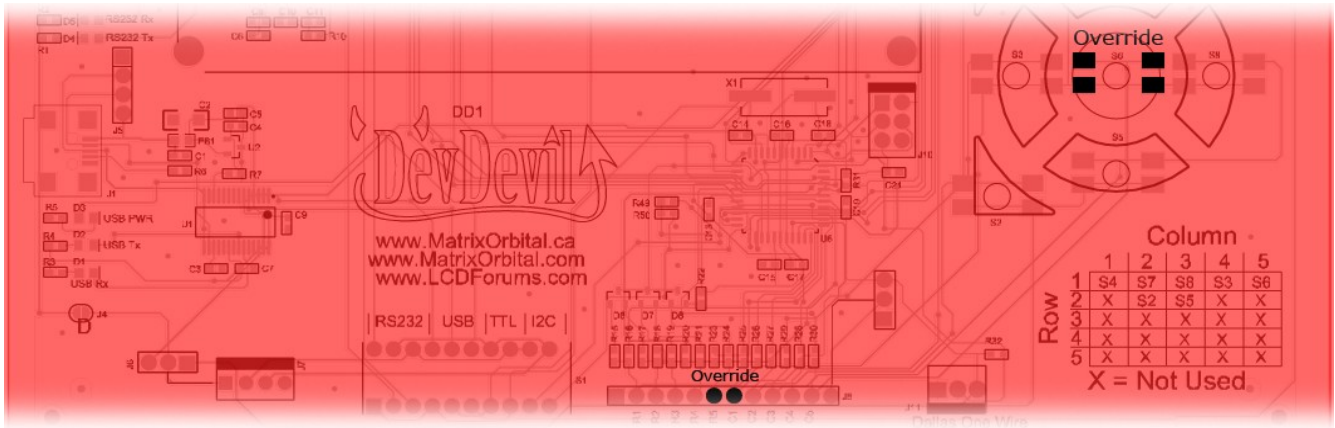


Illustration 16: Manual Override

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
I ² C 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 ² C 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			

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

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

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

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 ² C			

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