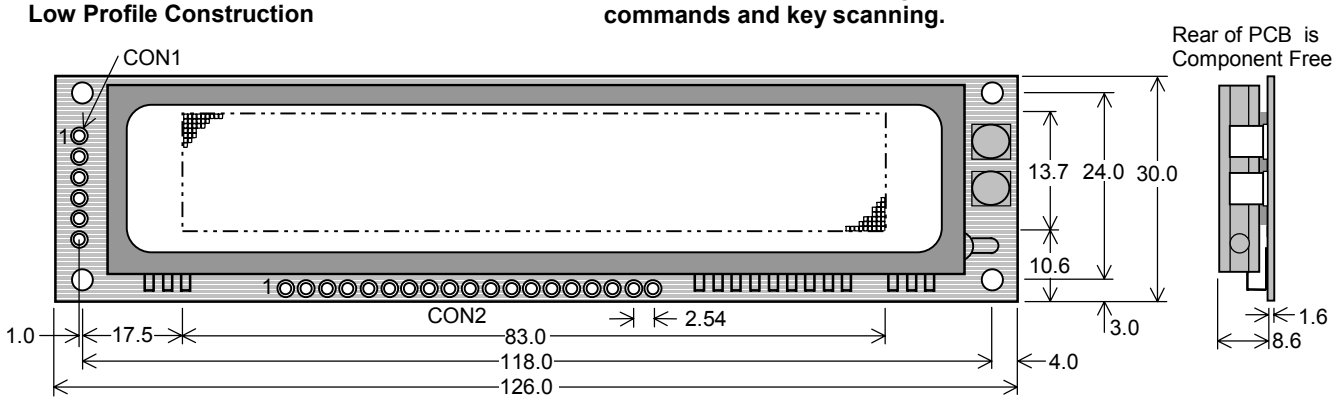
Dot Graphic VFD Module GU128x18F-K612A2

- **128 x 18 High Brightness Dot Graphic Display**
- **Single 5V DC Supply**
- **2 ASCII Fonts (5 x 7 and 10 x 14)**
- **Mixed Graphics and Text Display**
- **RS232 Serial Interface**
- **SPI Clock Serial Interface and 12 I/O Pins**
- **Variable Active Writing Area**
- **Low Profile Construction**

The module includes the VFD glass, VF drivers and microcontroller with refresh RAM, character generation, interface logic and patented transformerless DC/DC converter. The RS232 serial interface is suitable for connection to a host PC serial port and accepts baud rates up to 38,400 with or without parity.
The module features a low profile design with numerous custom options available including special fonts, application specific commands and key scanning.

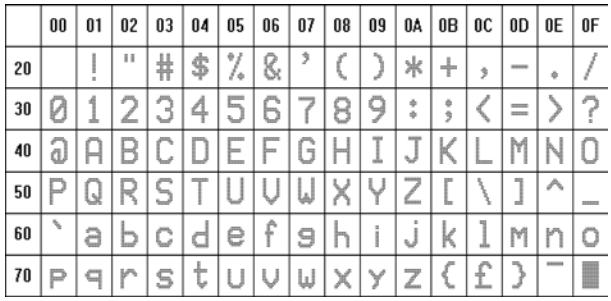


Dimensions in mm & subject to tolerances. Mounting holes 3.5mm dia.

ELECTRICAL SPECIFICATION

Parameter	Symbol	Value	Condition
Power Supply Voltage	V_{DD}	5.0VDC +/- 10%	GND=0V
Power Supply Current	I_{DD}	300 mA typ.	VDD=5VDC
RS232 Serial High Input	V_{IH}	3.0VDC min.	VDD=5VDC
RS232 Serial Low Input	V_{IL}	0.8VDC max.	VDD=5VDC
RS232 Serial High Output	V_{OH}	8.0VDC min.	VDD=5VDC
RS232 Serial Low Output	V_{OL}	-4.0VDC max.	VDD=5VDC
Logic High Input	V_{IH}	2.0VDC min.	VDD=5VDC
Logic Low Input	V_{IL}	0.8VDC max.	VDD=5VDC
Logic High Output	V_{OH}	2.4VDC min.	I_{OH}=-2.0mA
Logic Low Output	V_{OL}	0.5VDC max.	I_{OH}=2.0mA

CHARACTER FONT 10 x 14

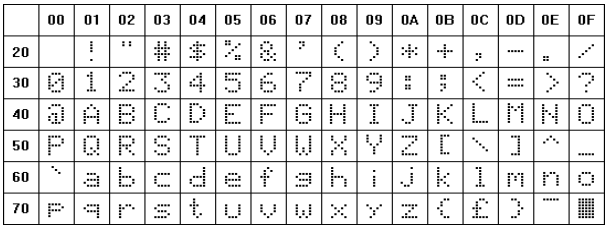


OPTICAL & ENVIRONMENTAL SPECIFICATION

Parameter	Value
Display Area (X xYmm)	83.05 x 13.69
Dot Size/Pitch (XxY mm)	0.5 x 0.6 / 0.65 x 0.77
Luminance	700 cd/m² (200 fL) Typ.
Colour of Illumination	Blue-Green (505nm)
Operating Temperature	-30°C to +80°C
Storage Temperature	-40°C to +85°C
Operating Humidity	20 to 85% RH @ 25°C

Optical filters can provide violet, red, yellow, blue & green output.

CHARACTER FONT 5 x 7



SOFTWARE COMMANDS

Command Name	Hex
Set Display Mode	10+
XON Handshake	11
Set I/O RS232, SPI, C0-3, D0-7	12+
XOFF Handshake	13
Set Active Area(llH,ttH,rrH,bbH)	14+
Position Cursor (lxH, tyH)	15+
Set Luminance (00H-1FH)	16+
Inverse Active Area	17
Fill Active Area	18
Outline Active Area	19
Write Graphic Bytes (len,nn..)	1A+
Extended Commands	1B+
Write ASCII Characters	20-7F

DISPLAY MODE SETTINGS (10H)

Bit	Function
1	0=Font 5x7 1=Font 10x14
2	0=Normal Font 1=Inverse Font
3	0=Cursor Increment 1=Dec.
4	0=Cursor Move Hor. 1=Vertical
5	0=Graphic Write Hor. 1=Vertical
6,7	Write mode 00=Overwrite 01=AND, 10=OR, 11=XOR

CON1

Pin	Signal	Function
1	EIN	RS232 Host Busy
2	EOUT	RS232 Module Busy
3	TXD	RS232 Transmit
4	GND	0V
5	RXD	RS232 Receive
6	VDD	5V

Link EIN and EOUT for XON/XOFF
Default: 19200, N, 8, 1

CON2

Pin	Signal	Function
1	GND	0V
2	/SS	Slave Select
3	RES	Reset Input
4	SIN	SPI Receive
5	SOUT	SPI Transmit
6	SCK	SPI Clock
7	GND	0V
8	VDD	5V
9-12	C3-C0	User I/O
13-20	D0-D7	User I/O

All I/O at Logic Level Voltages

Certain commands operate within the specified active area. The writing position (cursor) can be positioned to dot resolution within the active area. I/O control allows the user to set C0-C3, D0-D7 as individual input or output or universal keyboard matrix. When MOSI and MISO are linked at power on, C0-C3 and D0-D7 set the asynchronous serial baud rate and parity, which is saved in internal EEPROM. JP5 on the front of the module must be solder linked for the SPI interface to operate. Detailed specification, software commands and interface timing are available on request.

CONTACT

Noritake Sales Office Tel Nos
Nagoya Japan: +81 (0)52-561-9867
Canada: +1-416-291-2946
Chicago USA: +1-847-439-9020
Munchen (D): +49 (0)89-3214-290
Itron UK: +44 (0)1493 601144
Rest Europe: +49 (0)61-0520-9220
www.noritake-itron.com

Subject to change without notice.
IUK Doc. No. 03979 Iss.3 20Jan03

SOFTWARE COMMANDS

Instruction	Hex	Description
Display Mode *(BUSY = 350us)*	10H + n	Set display mode. Each bit in byte 'n' corresponds to a particular setting as follows:- Bit 1 Set font 0 = 5x7 , 1 = 10x14 Bit 2 Font mode 0 = Normal , 1 = Inverse Bit 3 Cursor position 0 = Increment , 1 = Decrement Bit 4 Cursor move 0 = Horizontal , 1 = Vertical Bit 5 Graphic write 0 = Horizontal , 1 = Vertical Bit 6,7 Write mode 00 = Overwrite , 01 = AND , 10 = OR , 11 = XOR *(The power-on default display mode is 00H).*
XON	11H	Asynchronous serial transmit enable (default). Ignored if using hardware handshaking, or synchronous serial communications.
Set I/O C3-C0 *(BUSY = 350us, 4ms [last byte])*	12H + 43H + n	Set port direction for user I/O lines C3–C0. 0=Output, 1=Input. (Bits 4 to 7 are ignored). This setting is stored in EEPROM. *Factory default is all inputs.*
Set I/O D7-D0 *(BUSY = 350us, 4ms [last byte])*	12H + 44H + n	Set port direction for user I/O lines D7–D0. 0=Output, 1=Input. This setting is stored in EEPROM. *Factory default is all inputs.*
Key Scan Mode *(BUSY = 350us, 4ms [last byte])*	12H + 46H	Key scan enable. Key codes are automatically sent to the asynchronous serial port. Key codes are in the form of: -4BH + n1H + n2H (n1H = C3-C0, n2H = D7-D0) This mode is stored in EEPROM.
I/O Mode *(BUSY = 350us, 4ms [last byte])*	12H + 47H	Disable key scanning and enable general I/O control. This mode is stored in EEPROM. *This mode is the factory default.*
Read Key *(BUSY = 350us)*	12H + 48H	Poll keyboard. Returns current key status. Key codes are in the form of: -4BH + n1H + n2H (n1H = C3-C0, n2H = D7-D0)
Read Inputs D7-D0 *(BUSY = 350us)*	12H + 54H	Request to read data from user I/O lines D7 – D0. Responds with:- 52H + nnH (nnH = D7-D0 pin status)
Read Inputs C3-C0 *(BUSY = 350us)*	12H + 55H	Request to read data from user I/O lines C3 – C0. Responds with:- 52H + nnH (nnH = C3-C0 pin status)
Set RS232 *(BUSY = 350us, 4ms [last byte])*	12H + 59H + n	Set baud rate and parity. Settings are stored in EEPROM and only take effect at power up.

Baud Rate	Bit 6	Bit 5	Bit 4	Parity	Bit 3	Bit 2	Handshaking	Bit 0
2400	0	1	0	NONE	0	0	Hardware	0
4800	0	1	1	ODD	0	1	Software	1
9600	1	0	0	EVEN	1	1		
1440	1	0	1					
19200	1	1	0					
38400	1	1	1					

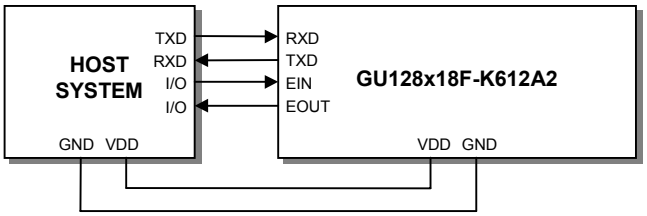
XOFF	13H	Asynchronous serial transmit disable. Ignored if using hardware handshaking, or synchronous serial communications.
Active Area *(BUSY = 350us)*	14H + left + top + right + bottom	Set active area. This defines the area in which all subsequent character and graphic writes occur. Current writing (cursor) position is always relative to the top left of the active area (0,0). Any data written that exceeds the active area is not displayed. *(The default is set to full display area).*
Cursor Position *(BUSY = 350us)*	15H + xpos + ypos	Position cursor. Sets the writing position within the current active area. *(default is 0,0)*
Luminance *(BUSY = 350us)*	16H + n	Set luminance. Set one of 32 levels of brightness. 00H Display off 01H Lowest brightness … 1FH Highest brightness *(default)*
Invert *(BUSY = 800us)*	17H	Active area invert. Note: Display mode has no effect on this command.
Fill *(BUSY = 800us)*	18H	Active area fill. Note: Display mode has no effect on this command.
Outline *(BUSY = 900us)*	19H	Draw box outline around active area. Note: Display mode has no effect on this command.
Graphic Write *(BUSY = 350us, 500us[data])*	1AH + len + data	Write graphic bytes. The format in which each byte is written and the action of the writing position between bytes is determined by values set using the set display mode command. len = number of graphical bytes to write.
Reset *(BUSY = 700us)*	1BH + 00H	Clear display and reset all settings to default values.
PSU Off *(BUSY = 350us)*	1BH + 01H	Switch internal PSU off. Display data is retained.
PSU On *(BUSY = 350us)*	1BH + 02H	Switch internal PSU on. *(Default)*
Set Outputs D7-D0 *(BUSY = 350us)*	1BH + 03H + n	Writes data byte n to user I/O ports D7–D0.
Set Outputs C3-C0 *(BUSY = 350us)*	1BH + 04H + n	Writes data byte n to user I/O ports C3–C0. (Bits 4-7 are ignored).
RS232 Out *(BUSY = 350us)*	1BH + 05H + n + 00H + 00H	Directs data byte n from synchronous serial port to asynchronous serial port. A status byte is return on the 5th byte (00H). Returned status byte: - 53H – successful 46H – failed.
Character Write *(BUSY = 600us)*	20H – 7FH	Write ASCII characters.

Jumper links : SPI enable link JP5.

INTERFACING TO THE GU128x18F-K612A2

The VFD module has two serial ports for communication. The synchronous serial port uses the MOSI, MISO and SCK lines and are at TTL level. The asynchronous serial port uses lines RXD, TXD, EIN and EOUT are at RS232 levels.

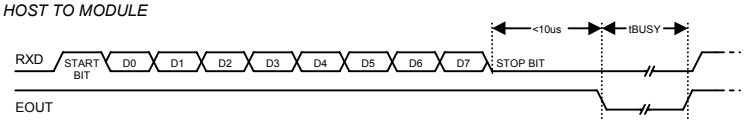
ASYNCHRONOUS SERIAL COMMUNICATION



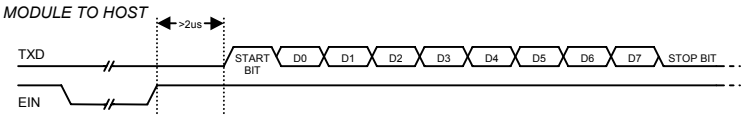
The communication speed and parity can be changed with the 'Set RS232' Software Command. These setting will be applied at power-up and with hardware reset. Linking 'EIN' and 'EOUT' will enable software handshaking. The factory default settings are 19200 baud with no parity and hardware handshaking.

Hardware handshaking is available using the module busy 'EOUT' and host busy 'EIN' control lines. The module contains a receive buffer of 16 bytes.

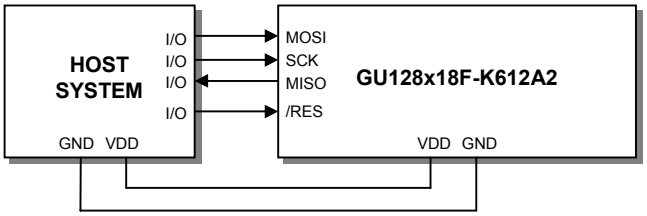
HOST TO MODULE



MODULE TO HOST

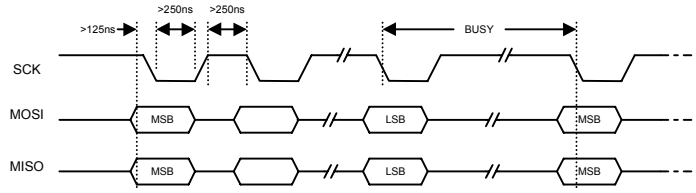


SYNCHRONOUS SERIAL COMMUNICATION

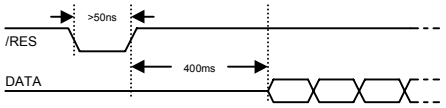


The host must provide adequate delays for the module to process the command and data. These 'busy' times are specified in the software command section on page 1.

When using synchronous communications, the data (on MOSI) is clocked in on the falling edge of SCK. The most significant bit of the data byte should be sent first. Please note that JP5 (back of module) must be linked to enable synchronous serial communication.



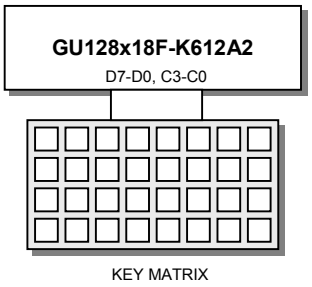
RESET TIMING



The module is reset when a low level signal is applied to the /RES line. The internal EEPROM will be checked to setup the I/O mode and the synchronous serial communications. During this initialisation period, the user must delay any transmission to the module.

KEY SCANNING

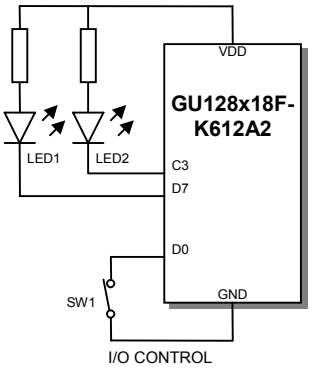
Key scanning is achieved using the I/O lines C3-C0 and D7-D0. The module will continuously scan these lines and check for any direct links. Up to 32 keys can be scanned. Once a link has been detected, the keyboard line status will be automatically sent to the host through the asynchronous interface. For synchronous serial communications, the host must poll the keyboard status. Please note, that the VFD module does not decode the keyboard, it only shows the column and row where a key press has been detected. This allows for many different key matrix configurations to be used, e.g. 4x8, 5x7, 6x6 etc.



Example key scanning: -

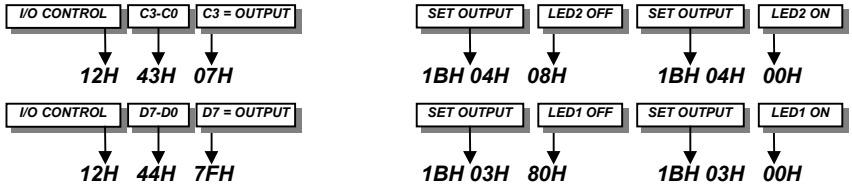


I/O CONTROL

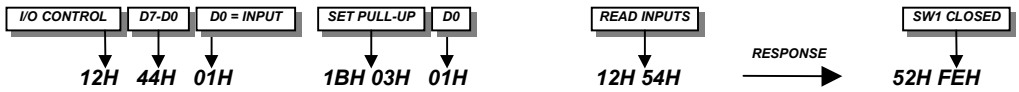


The module contains simple Input and Output functions for the 12 I/O lines (D7-D0, C3-C0). All inputs include an optional pull-up resistor, 30K-120K in value. The outputs can source ~5mA and sink ~30mA.

The following example sets up the I/O lines to control the 2 LED's.



The next example checks to see if the switch has been closed.



ACTIVE AREA

All data written to the display will reside in the 'Active Area'; this area is, by default, set to the whole display (128x18 pixels). The user can however, change the size and position of this area. All cursor positioning will be relative to the top left corner of the 'Active Area'. The active area can be used for block inversion & filling, and drawing box outlines. It can also be used for constraining text and graphical data to any part of the display.

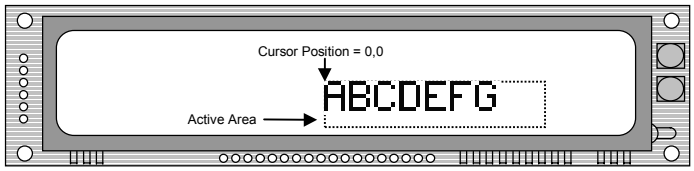
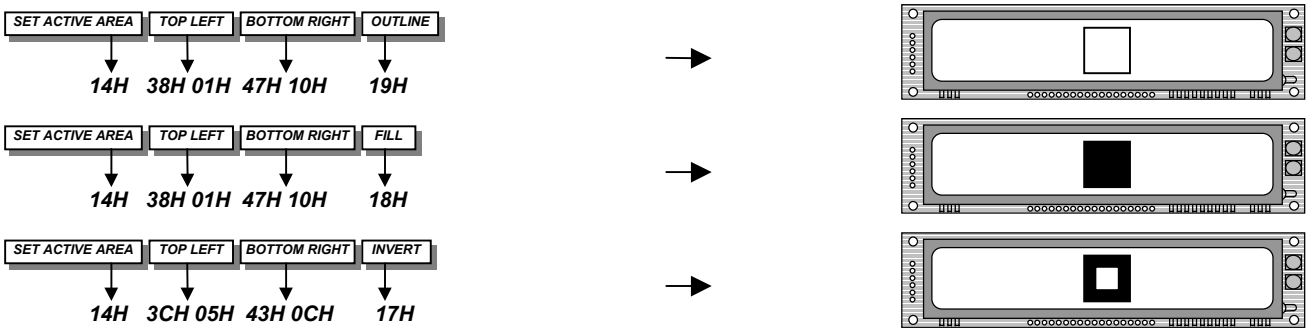


Fig2: Active Area and cursor positioning.

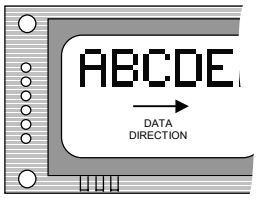
The following examples show some simple active area commands: -



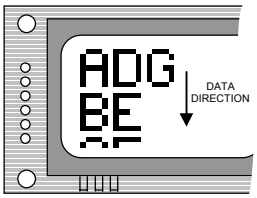
CURSOR MOVEMENT

The cursor is automatically repositioned after each data write and the direction can be selected by the 'Display Mode' command. The 'Cursor Move' bit selects either an up/down or left/right direction. The 'Cursor Position' bit selects either a forward or backward movement. The diagrams below show how the data is written to the display with different cursor movements.

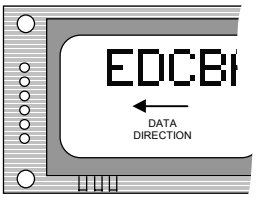
CHARACTER WRITES



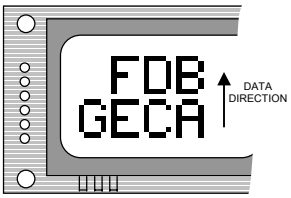
Cursor Movement = Increment
Cursor Move = Horizontal



Cursor Movement = Increment
Cursor Move = Vertical

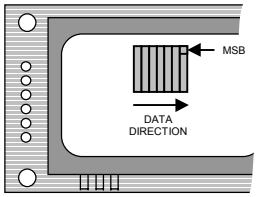


Cursor Movement = Decrement
Cursor Move = Horizontal

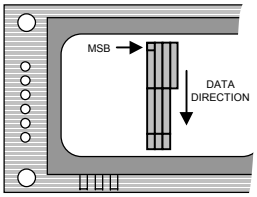


Cursor Movement = Decrement
Cursor Move = Vertical

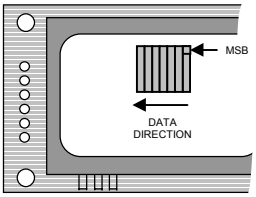
GRAPHIC WRITES (Vertical Data)



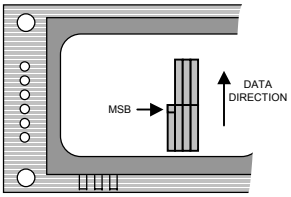
Cursor Movement = Increment
Cursor Move = Horizontal



Cursor Movement = Increment
Cursor Move = Vertical

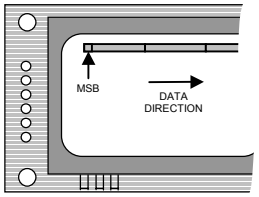


Cursor Movement = Decrement
Cursor Move = Horizontal

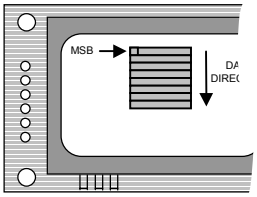


Cursor Movement = Decrement
Cursor Move = Vertical

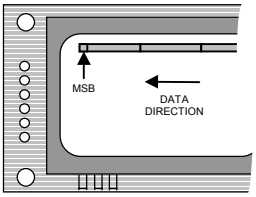
GRAPHIC WRITES (Horizontal Data)



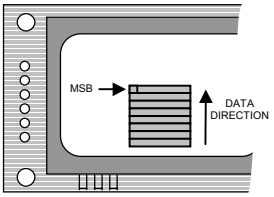
Cursor Movement = Increment
Cursor Move = Horizontal



Cursor Movement = Increment
Cursor Move = Vertical



Cursor Movement = Decrement
Cursor Move = Horizontal



Cursor Movement = Decrement
Cursor Move = Vertical

DISPLAYING TEXT

The module contains 2 font sizes, a 5x7 pixel and a 10x14 pixel font. Characters of either size can be written to any part of the display. All data sent to the module from 20H to 7FH is treated as character data. Characters are positioned below the current cursor position, see Fig1. Each character written will include a one pixel space to the right and below. After each character is written to the display, the cursor position is automatically advanced, see the cursor movement diagrams below. By default the cursor position 0,0 defines the top-left of the display, this can however be changed when using the 'Active Area' command. This is explained in more detail on the previous page.

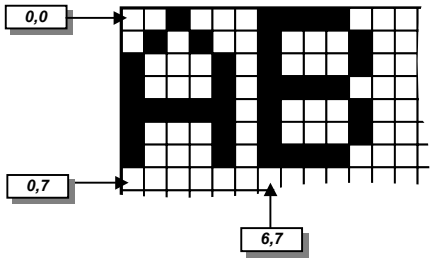
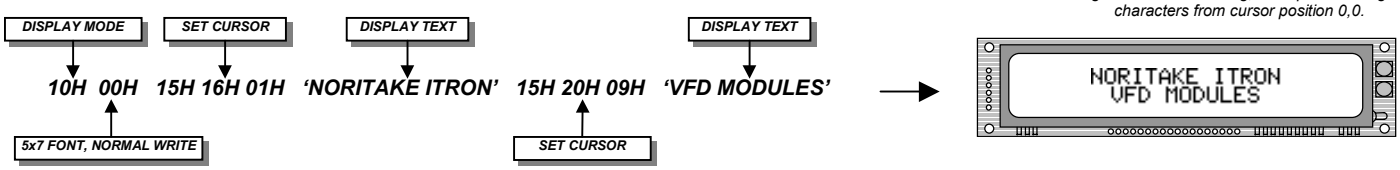
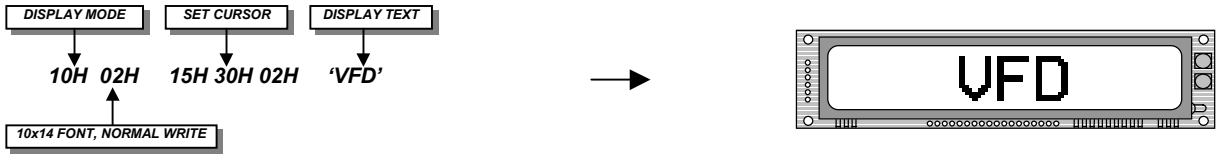


Fig1: Cursor Positioning, example of writing 2 characters from cursor position 0,0.

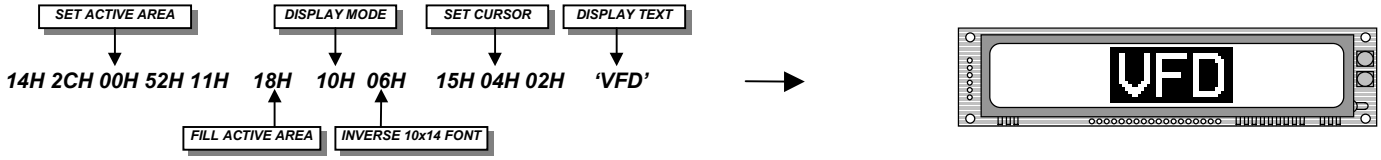
The following example displays two text messages in the center of the display.



The next example displays one line of text using the 10x14 font.



This example displays inverted text on a filled area.



DISPLAYING GRAPHICS

Graphical images can be displayed on the VFD module in either a horizontal or vertical byte orientation. After each graphical data write, the cursor is automatically advanced, depending upon the direction selected in the 'Display Mode' command. The most significant bit is positioned to the top (vertical data) or to the left (horizontal data).

The following example displays a simple graphic image. The vertical graphics command is used to send the top 20 bytes, then the bottom 20 bytes of graphical data.

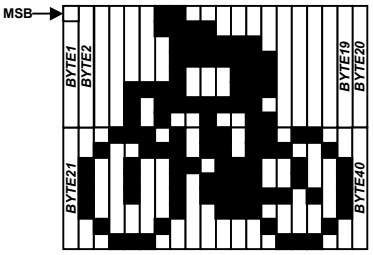
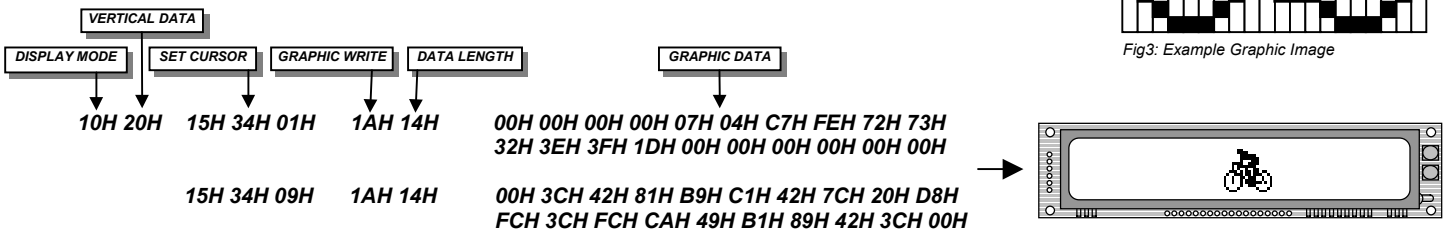


Fig3: Example Graphic Image



WRITE MODES

By default, display data that is overwritten will be cleared prior to displaying any new data. The display data can be maintained whilst writing new data by setting the 'OR' bits in the 'Display Mode' command, this will effectively merge the old data with the new. Other write modes include 'AND' and 'XOR'. The 'AND' write mode will only display data where there is previously written data, and the 'XOR' write mode inverts all overwritten data.