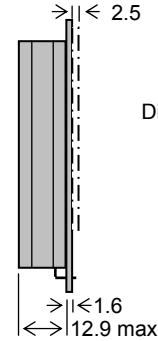
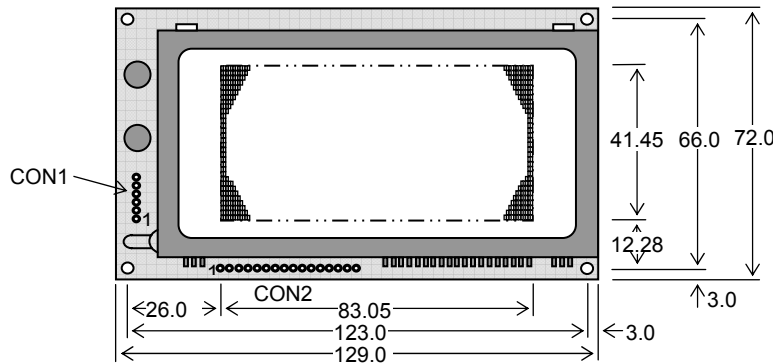


Dot Graphic VFD Module

GU126x64F-K612A4

- 126 x 64 High Brightness Dot Graphic Display
- Single 5V DC Supply
- 3 ASCII Fonts (5 x 5, 5 x 7, and 10 x 14)
- SPI & RS232 Asynchronous Serial Interface
- 8 User I/O Pins with Key Scanning Capability
- Transformerless PSU (patent pending)
- Low Profile Construction

The module includes the VFD glass, VF drivers and micro-controller with refresh RAM, character generation, interface logic and patented transformerless DC/DC converter. The RS232 serial interface is full duplex and accepts various baud rates up to 38,400. The module features a low profile design with numerous custom options available including special fonts and application specific commands.



Dimensions in mm & subject to tolerances.
Mounting holes 3.2mm dia.

ELECTRICAL SPECIFICATION

| Parameter | Symbol | Value | Condition |
|----------------------|-------------------------------------|-------------------------|-------------------------|
| Power Supply Voltage | V _{DD} | 5.0VDC +/- 5% | GND=0V |
| Power Supply Current | I _{DD} | 750 mA typ. | V _{DD} =5VDC |
| RS232 Input | V _{SIL} / V _{SIH} | -24V max / +24V max | V _{DD} =5VDC |
| RS232 Output | V _{SOL} / V _{SOH} | -5VDC min / +5VDC min | V _{DD} =5VDC |
| Logic Input | V _{IL} / V _{IH} | 0.8VDC max / 2.0VDC min | V _{DD} =5VDC |
| Logic Output | V _{OL} / V _{OH} | 0.5VDC max / 2.4VDC min | I _{OH} =-2.0mA |

OPTICAL & ENVIRONMENTAL SPECIFICATION

| Parameter | Value |
|-------------------------|-------------------------------------|
| Display Area (X x Ymm) | 83.05 x 41.45 |
| Dot Size/Pitch (XxY mm) | 0.5 x 0.5 / 0.65 x 0.65 |
| Luminance | 600 cd/m ² (200 fL) Typ. |
| Colour of Illumination | Blue-Green (505nm) |
| Operating Temperature | -40°C to +85°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.

SOFTWARE COMMANDS

| Hex | Command | Hex | Command |
|-----------------|-------------------|----------------------|---------------------------------|
| 01-07 | Run Macro | 19 | Reset |
| 08 | Backspace | 1A + data | Write Mode |
| 09 | Horizontal Tab | 1B + macro+len+data | Set Macro |
| 0A | Line Feed | 1B + 4D | Erase All Macros |
| 0B | Home | 1B + 4C/55 | Lock/Unlock EEPROM |
| 0C | Vertical Tab | 1B + 43 | Request Checksum |
| 0D | Carriage Return | 1B + 50/46 | Power On/Off |
| 0E | Clear End of Line | 1B + 48/42 | Hex Write On/Off |
| 0F | Test | 1B + 49 + data | Set Comms |
| 10 + x + y | Cursor Position | 1B + 44 + data | Enable I/O Port |
| 11 +xl+yt+xr+yb | Set Area | 1B + 4F + data | Set Port Lines |
| 12 +xl+yt+xr+yb | Clear Area | 1B + 52 | Read Port |
| 13 +xl+yt+xr+yb | Invert Area | 1B + 4B | Enable key scanning |
| 14 +xl+yt+xr+yb | Set Outline | 1B + F8-FF | Brightness |
| 15 +xl+yt+xr+yb | Clear Outline | 1C / 1D / 1E | Select Font |
| 16 | Set Pixel | 1F +xl+yl+xr+yb+data | Graphic Area Write ¹ |
| 17 | Clear Pixel | 20 - 7F | Character Write |
| 18 | Graphic Write | | |

¹ Applies to version 3 software only.

The module defaults to an 8 line of 21 character display using the 5x7 font with single pixel spacing. The cursor position auto increments after each character write. The bottom left of a character is placed at the cursor x,y. To send commands as hexadecimal, prefix the 2 bytes using character 60H. Example: `10'3F'01 = Position dot x=63 y=1. To send character 60H to the display, send 60H twice.

CHARACTER SETS

MINI FONT (PROPORTIONAL SPACING)

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 20 | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / | |
| 30 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 40 | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | |
| 50 | p | q | r | s | t | u | v | w | x | y | z | [| \ |] | ^ | _ |

5x7 & 10x14 FONTS (FIXED SPACING)

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 20 | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / | |
| 30 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 40 | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | |
| 50 | p | q | r | s | t | u | v | w | x | y | z | [| \ |] | ^ | _ |
| 60 | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 70 | p | q | r | s | t | u | v | w | x | y | z | [| \ |] | ^ | _ |

CON1

| Pin | Signal |
|-----|-----------------|
| 1 | VDD |
| 2 | RXD (RS232 in) |
| 3 | 0V |
| 4 | TXD (RS232 out) |
| 5 | MB (RS232 out) |
| 6 | HB (RS232 in) |

CON2

| Pin | Signal | Pin | Signal |
|-----|--------|------|--------|
| 1 | /SS | 6 | MISO |
| 2 | 0V | 7 | SCK |
| 3 | VDD | 8 | /IRQ |
| 4 | /RES | 9-16 | P0-P7 |
| 5 | MOSI | | |

The Module Busy line (MB) indicates the module is busy when low. If handshaking is not required, connect the Host Busy (HB) input to the Module Busy output.

Connect MISO and MOSI at power-up to enable the test mode and restore factory defaults.

CONTACT

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

Subject to change without notice.
IUK Doc. No. 003943 Iss.4
25 Oct 06

Dot Graphic VFD Module

GU126x64F-K612A4

SOFTWARE COMMANDS

| Instruction | Data Format | Description |
|---|------------------------------------|--|
| Macro Start (BUSY time depends on contents) | 01H - 07H | Start user defined macro 1-7. |
| Backspace (50µs) | 08H | Non destructive backspace. Cursor is moved left by the width of the currently select font. If the cursor is at the left end of the display, no cursor movement is made. |
| Horizontal Tab (50µs) | 09H | Cursor is moved right by the width of the currently select font. If the cursor is at the end of the display, no cursor movement is made. |
| Line Feed (50µs) | 0AH | Moves the cursor down by the height of the currently selected font. If the cursor is at the bottom of the display, no cursor movement is made. |
| Home (50µs) | 0BH | Moves the cursor horizontal position to 00H, the vertical positioning is dependent on the currently selected font, allowing for immediate character writing in the top-left corner of the display. |
| Vertical Tab (50µs) | 0CH | Moves the cursor up one character row. If the cursor is at the top of the top end of the display, no cursor movement is made. |
| Carriage Return (50µs) | 0DH | Moves the cursor horizontal position to 00H. The vertical position is unchanged. |
| Clear EOL (3ms) | 0EH | Clear all characters from the current cursor position to the end of the display. |
| Test (50µs) | 0FH | Place module into self-test mode. The module will repetitively show a few test screens. The test mode will exist on the next received byte. |
| Cursor Position (50µs) | 10H + xpos + ypos | Set the cursor position. |
| Set Area (50µs + 2ms [last byte]) | 11H + xleft + ytop + xright + ybot | Fill specified area. All dots within the specified area are illuminated. Please note that the cursor position is affected with this command. |
| Clear Area (50µs + 2ms [last byte]) | 12H + xleft + ytop + xright + ybot | Clear specified area. All dots within the specified area are cleared. Please note that the cursor position is affected with this command. |
| Invert Area (50µs + 2ms [last byte]) | 13H + xleft + ytop + xright + ybot | Invert specified area. All dots within the specified area are inverted. Please note that the cursor position is affected with this command. |
| Set Outline (50µs + 2ms [last byte]) | 14H + xleft + ytop + xright + ybot | Draw box outline. All dots within the specified outline are unchanged. Please note that the cursor position is affected with this command. |
| Clear Outline (50µs + 2ms [last byte]) | 15H + xleft + ytop + xright + ybot | Clear box outline. All dots within the specified outline are unchanged. Please note that the cursor position is affected with this command. |
| Set Pixel (50µs) | 16H | Illuminate a single pixel at the current cursor position. |
| Clear Pixel (50µs) | 17H | Clear a single pixel at the current cursor position. |
| Graphic Write (50µs + 350µs [each data byte]) | 18H + len + data | Write graphical data, length <i>len</i> , direct to display. See write mode command (1AH) for graphic orientation and cursor movements. |
| Reset (1ms) | 19H | Resets display to power-on defaults: - Display is cleared. 5x7 font selected. Write Mode = 00H Brightness Level = 7. VFD Power = On. |
| Write Mode (50µs) | 1AH + data | Bit 7 = graphic data orientation - 0 = horizontal, 1 = vertical (<i>default = horizontal</i>) Bit 6 = cursor movement - 0 = horizontal, 1 = vertical (<i>default = horizontal</i>) Bit 5 = cursor direction - 0 = forward, 1 = backwards (<i>default = forwards</i>) Bit 4 = underscore cursor - 0 = off, 1 = on (<i>default = off</i>) Bit 3 = underscore cursor - 0 = static, 1 = flash (<i>default = static</i>) Bit 1/0 = pen type - 00 = overwrite, 01 = AND, 02 = OR, 03 = XOR (<i>default = overwrite</i>) |
| Set Macro (50µs [1BH], 10ms [macro], 50µs [len], 5ms [data], 20ms [last byte]) | 1BH + macro + len + data | Send macro data to EEPROM. <i>macro</i> = 00H - 07H. Macro0 is executed at power-up only. A maximum of 480 bytes is allowed for macro data. The display may flicker whilst writing macro data. |
| Brightness (50µs) | 1BH + level | Set the display brightness. level = F8H - FFH. F8H = display off. F9H = minimum, FFH = maximum (<i>default</i>). |
| Erase Macros (100ms) | 1BH + 4DH | Clear all downloaded macros in EEPROM. Screen may blank momentarily while macro data is being erased. |
| Lock/Unlock EEPROM (50µs + 5ms [last byte]) | 1BH + 4CH / 55H | All data contained within the non-volatile EEPROM is locked (4CH), and no changes are possible until the unlock command (55H) is executed. |
| Checksum (50µs) | 1BH + 43H | All data received is added to the checksum. This command will read the lower 8-bits of that checksum, before being cleared. Please note that the checksum is cleared when executing the test mode. |
| Power On/Off (50µs) | 1BH + 50H / 46H | 50H = Turn on VFD power supply (<i>default</i>). 46H = Turn off VFD power supply, display's contents will be preserved. |
| Hex/Binary Mode (50µs) | 1BH + 48H / 42H | 48H = Enable hex receive mode, character 60H is interpreted as a hexadecimal prefix. 42H = Disable hex receive mode. Hex mode is enabled at power up. |
| Set Comms (50µs + 5ms [last byte]) | 1BH + 49H + data | Set asynchronous communication baud rate and parity. Takes effect at power-up or hardware reset. Bit 7 = Automatic I/O send (0=off, 1=on). Bits 1&0 = baud rate (00=4800, 01=9600, 02=19200, 03=38400). Bit 2 = Parity (1=even, 0=none) (<i>factory default = 19200 with no parity, automatic I/O send is off</i>) |
| Enable I/O Port (50µs + 5ms [last byte]) | 1BH + 44H + data | Set I/O port direction. A '1' indicates an input, a '0' an output. All output lines are immediately set low. All input lines have their pull-ups enabled. This value is store in EEPROM and will automatically be set at power up. |
| Set Port Lines (50µs) | 1BH + 4FH + data | Set Output lines on I/O port, a '1' will set 5V on the output ports, or enable the pull-ups on the inputs. |
| Read Port (50µs) | 1BH + 52H | Read current I/O port status. A single byte is transmitted showing the current state of the I/O lines. |
| Enable Key Scanning (50µs + 5ms [last byte]) | 1BH + 4BH | Set I/O port to key scanning. The I/O ports are continuously scanned for any key press. This mode is stored in EEPROM and will automatically be selected at power up. |
| Select Font (50µs) | 1CH / 1DH / 1EH | Select font. 1CH = proportional mini font. 1DH = fixed spaced 5x7 font. 1EH = fixed spaced 10x14 font. |
| Graphic Area Write (50µs + 350µs [each data byte]) | 1FH + xl + yt + xr + yb + data | Write graphic data within defined area. See write mode command (1AH) for graphic orientation and cursor movements. Note: This command is available on software version 3 only. |
| Hex Prefix (50µs + 50µs + command BUSY) | 60H + dhH + diH | Write to the display module using a 2-byte hexadecimal number. dhH = high nibble, diH = low nibble. E.g. Sending '19 will reset the display. |
| Character Write (800µs) | 20H - 7FH | Display character from selected font. |

Notes: - Busy times are not inclusive of a 100µs scan period, this must be taken into consideration. If the cursor is enabled, busy times will increase by a further 50µs. All coordinates are absolute. The origin (00H, 00H) is the top left of the display. All data shown is in hexadecimal format.

INTERFACING TO THE GU126x64F-K612A4

All control communication to the VFD module is by the TTL asynchronous or synchronous serial interfaces. The RS232 asynchronous communication speed and parity can be changed with the 'SET COMMS' command. These settings are stored in EEPROM and become active at power up or hardware reset. The default settings are 19200 baud, with no parity.

ASYNCHRONOUS SERIAL COMMUNICATION

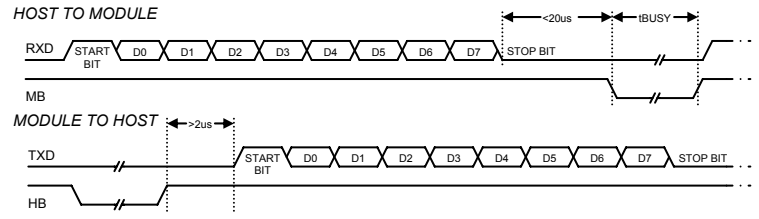
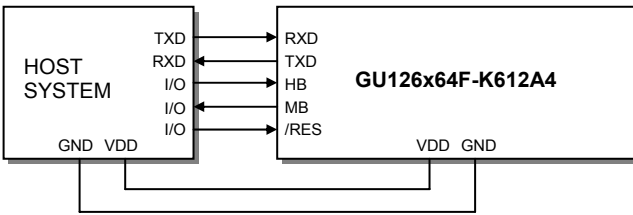
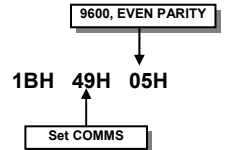
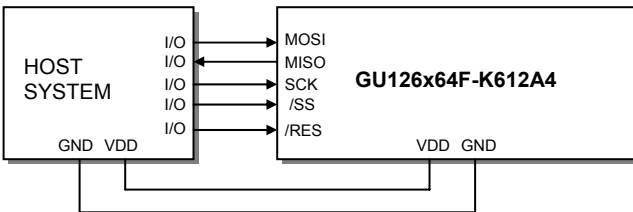


Fig1: Waveforms show RS232 asynchronous serial.

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. To change from the default communication settings, use the 'SET COMMS' command. Any changes take effect on the next hardware reset or power up. Settings are stored in non-volatile EEPROM. The example on the right shows how to change the baud rate and parity.



SYNCHRONOUS SERIAL COMMUNICATION



When using synchronous communications. Data is clocked in on the rising edge of SCK. The most significant bit of the data byte should be sent first.

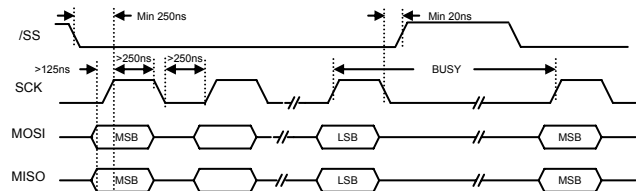
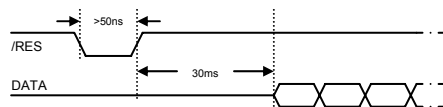


Fig2: Waveforms show TTL synchronous serial.

The synchronous serial interface is active and able to receive data when /SS is held low. The /SS line is useful for packet / byte synchronization to keep the internal bit counter logic synchronous with the host. When /SS is driven high the send and receive logic is reset and any partially received data is discarded. The host must provide adequate delays for the module to process the data. These data/command busy times are specified in the software command section.

RESET TIMING

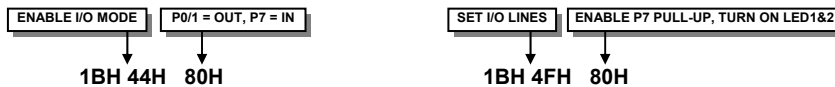


The module is reset when a low-level signal is applied to the /RES line. This will cause the module to clear the display, initialise the UART setting and set all power-up defaults. During this initialisation period, the user must delay any transmission to the module.

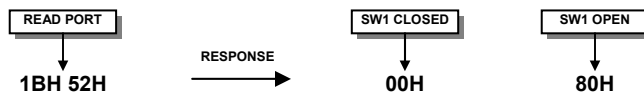
I/O CONTROL

The module contains simple Input and Output functions for the 8 I/O lines (P7-P0). All inputs include an optional pull-up resistor, 30K-120K in value. The outputs can source ~5mA and sink ~30mA.

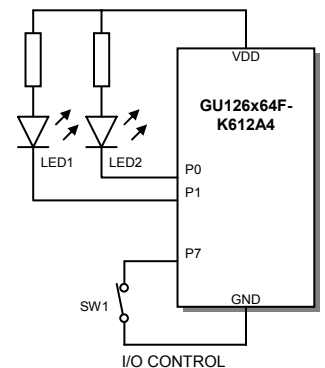
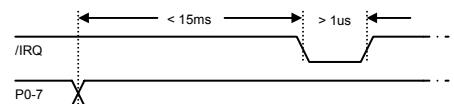
The following example sets up the I/O lines to control the 2 LED's and provide a pull-up resistor for the switch.



The status of P0-P7 can be transmitted when a change in level is detected on any pin. This automatic response mode can be enabled by using the 'UART SETUP' command. When this mode is enabled, the VFD module can reliably check port changes every 15ms. With auto send disabled (default) a manual read command is required to determine the port status.



When using synchronous serial communications, the host can detect a level change with the /IRQ line on CON2. This allows the host to poll the port status only when needed.



DISPLAYING TEXT

The module contains 3 font sizes, a proportional mini-font, 5x7 pixel, and a 10x14 pixel font. Characters of any 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 above the current cursor position, see Fig1. Each character written will include a one pixel space on the right side of the character. After each character is written to the display, the cursor position is automatically advanced. If the cursor position reaches the end of the display, the host must reposition to the next line.

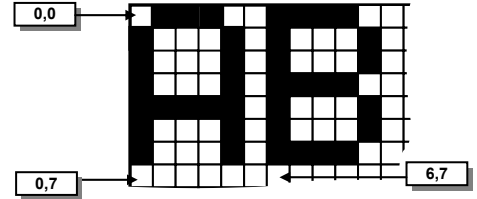
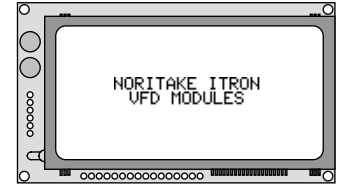
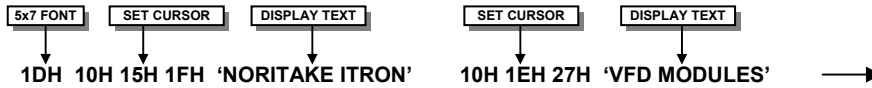


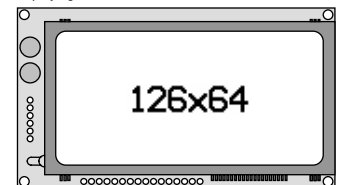
Fig3: Cursor Positioning, example of writing 2 characters from cursor position 0,7.

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



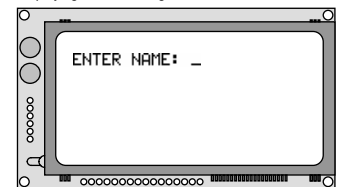
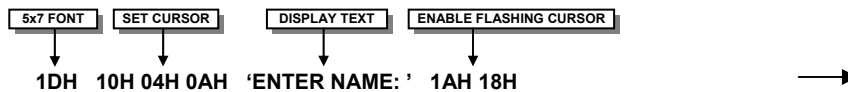
Displaying text in the small 5x7 font.

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



Displaying text in the large 10x14 font.

The module can display a cursor to aid character positioning and text input. The size of the cursor depends upon the currently selected font, and can be set to flash or remain static.



Using the cursor to aid user input.

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 'Write 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 graphical image using horizontal graphic data. The write mode is first set to horizontal data format, with a vertical cursor movement. The cursor is positioned before sending the 8 byte of graphical data using the graphics command.

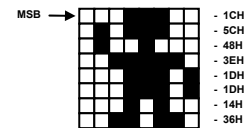
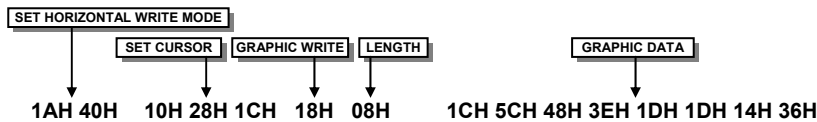


Fig4: Graphic Image using horizontal data

The next example displays a simple graphical image using vertical graphical data. The write mode is first set to vertical data format, with a horizontal cursor movement. The cursor is positioned, then the top 20 bytes are sent using the graphic write command. The cursor is then repositioned to send the bottom 20 graphical bytes.

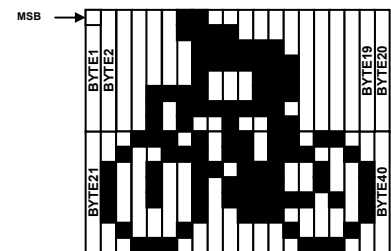
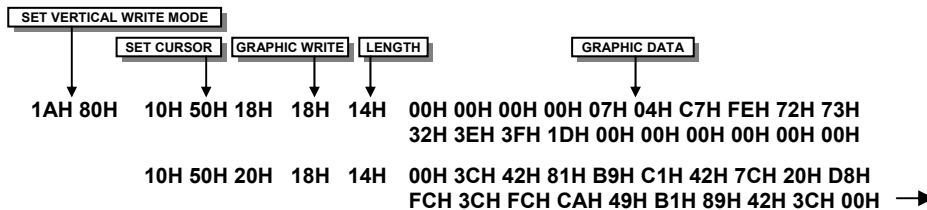
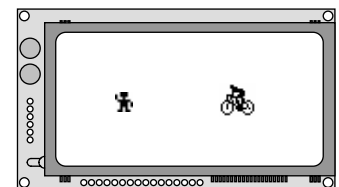


Fig5: Graphic Image using vertical data



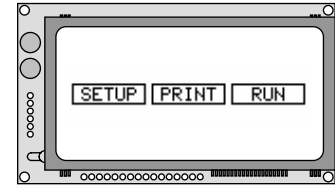
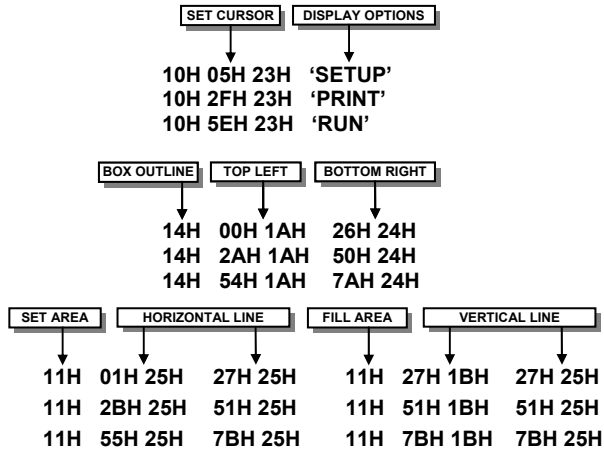
Displaying graphic images in vertical and horizontal format.

The graphic area write command 1FH uses top-left and bottom-right XY co-ordinates to define an area to which graphical data bytes will be written. The orientation is set-up using the write mode command 1AH. Unused bits are masked where the screen area is not a byte multiple.

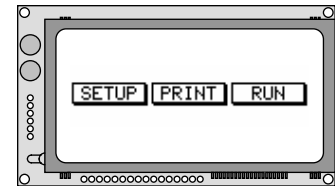
AREA COMMANDS

The VFD module contains commands to fill, clear and invert defined areas of the display. Also an outline command is available to draw rectangles around objects.

The following example displays three options for the user to select, each option is contained within a box with a shadow effect. Drawing horizontal and vertical line using the fill area command creates the shadow effect.

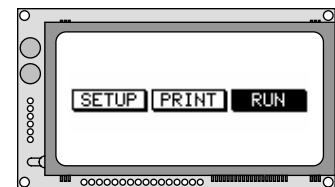


Boxes created using the 'Set Outline' command.



Drop Shadows created with the 'Set Area' command.

The next example uses the invert area command to select one of the options.

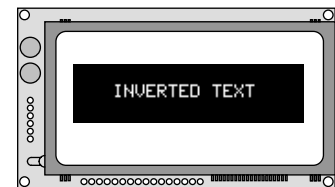
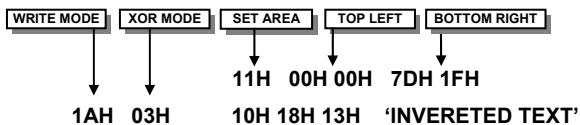


Option 'Run' selected with the 'Invert Area' command.

WRITE MODES

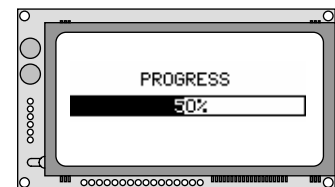
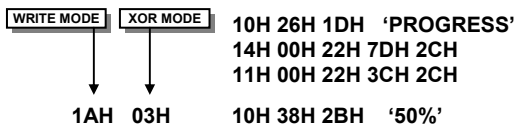
By default, display data that is overwritten will be cleared prior to displaying any new data. This display data can be maintained whilst writing by selecting the 'OR' mode with the 'Write Mode' command, this will effectively merge the old data with the new. The 'AND' write mode will only display written data if existing data is present on the display. The other 'Write Mode' is 'XOR' which can be useful for writing text on an inverted display.

The following example uses the XOR mode to write text on a full display.



Displaying inverted text using the 'Write Mode' command.

This next example uses the XOR mode to display the percentage completed on a progress bar.



Using inverting text for displaying progress level.

MACROS

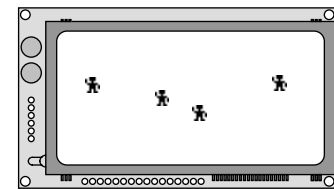
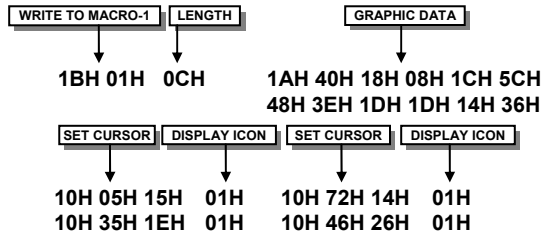
A string of data and commands can be sent to the module and stored in non-volatile EEPROM by using the macro feature. This string of data and commands can then be executed by using just one command. Up to 8 macros can be used at any one time, one of these is executed at power-up.

This example uses the first macro (Macro 0) to display an initial message at power-up.



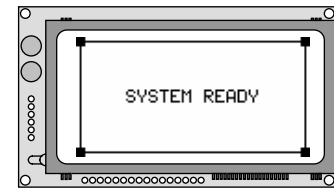
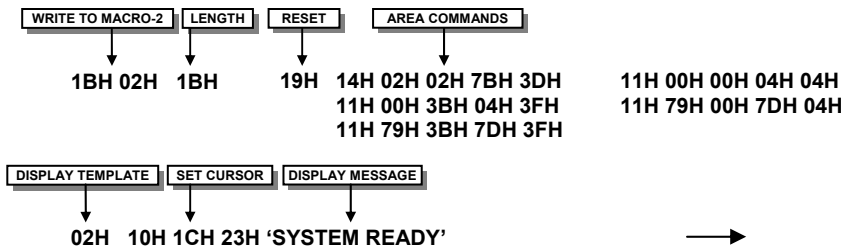
Power-Up message using Macro 0.

This next example saves the previous graphic icon into Macro 1, and then is used as a user-defined character.



Using Macros as user-defined characters.

This example creates a display template, which can be helpful if many screens require the same look.



Using Macros as a screen template.

EEPROM PROTECTION

The EEPROM contains information such as macro data, asynchronous communication settings and I/O configuration. So it is important to protect this information from stray commands due to communication failures. To protect the EEPROM, the module contains a 'EEPROM Lock' command (1BH + 4CH). Once this command is issued, no further EEPROM updates can be made until it is unlocked (1BH + 55H).

USING THE CHECKSUM

All data written to the module is added to an internal checksum. The lower 8-bits of this checksum can be read at any time from the module by the host system to confirm accurate data transfer. It is up to the user if or when this feature should be used. The checksum is cleared at power-up and after each checksum read.

Example: Read checksum at power-up, or directly after it has been cleared.



Example: Read checksum after data has been written to the display.

