



**Specification:**            **GU180X32D-K610A8**  
                                     **VFD Module**

**Features:**                **180x32 High Brightness Dot Graphic Display**  
                                     **Single 5V DC Supply**  
                                     **4 ASCII Fonts ( 5x5, 5x7, 10x14, 7x15 )**  
                                     **Asynchronous, SPI, I2C & Parallel interfaces**  
                                     **8 User I/O Pins with Key Scanning Capability**

**Reference:**                40112  
**Issue Date:**            29th September 2008  
**Revision:**                F3

**[www.noritake-itron.com](http://www.noritake-itron.com)**

This specification is subject to change without prior notice.

Index		Page
<b>1.0</b>	<b>General Description</b>	3
	1.1 Scope	3
	1.2 Construction	3
	1.3 Outline	3
	1.4 Block Diagram	3
<b>2.0</b>	<b>Electrical Specification</b>	4
	2.1 Absolute Maximum Ratings	4
	2.2 Electrical Ratings	4
	2.3 Electrical Characteristics	4
<b>3.0</b>	<b>Optical Specifications</b>	5
<b>4.0</b>	<b>Environmental Specifications</b>	5
<b>5.0</b>	<b>Interface</b>	5
	5.1 Type of Interface	5
	5.2 Parallel Interface	5
	5.2.1 Basic Function	5
	5.2.2 Parallel Interface Timing	6
	5.3 Serial Interface	6
	5.3.1 Basic Function	6
	5.3.2 Asynchronous Serial Interface Timing	6
	5.3.3 Synchronous Serial Interface Timing	7
	5.3.4 I2C Serial Interface Timing	8
	5.4 Reset	8
	5.5 Serial Buffer	9
	5.6 Packet Mode	9
	5.7 EEPROM Protection	9
	5.8 Keyboard Control	10
	5.9 I/O Control	11
<b>6.0</b>	<b>Software Commands</b>	12
	6.1 Dummy Byte	12
	6.2 Macro Start	12
	6.3 Backspace	12
	6.4 Horizontal Tab	12
	6.5 Line Feed	12
	6.6 Home	12
	6.7 Vertical Tab	12
	6.8 Carriage Return	12
	6.9 Clear EOL	12
	6.10 Test	12
	6.11 Cursor Position	13
	6.12 Set Area	13
	6.13 Clear Area	13
	6.14 Invert Area	13
	6.15 Set Outline	13
	6.16 Clear Outline	13
	6.17 Set Pixel	13
	6.18 Clear Pixel	13
	6.19 Graphic Write	13
	6.20 Reset	13
	6.21 Write Mode	14
	6.22 Set Macro	14
	6.23 Brightness	14
	6.24 Erase Macros	14
	6.25 Lock/Unlock EEPROM	14
	6.26 Checksum	14
	6.27 Power On/Off	14
	6.28 Hex/Binary Mode	14

Index		Page
	6.29 Set Serial Communications	15
	6.30 Enable I/O Port	15
	6.31 Set Port Lines	15
	6.32 Read Port	15
	6.33 Enable Key Scanning	15
	6.34 Select Font	15
	6.35 Graphic Area Write	15
	6.36 Character Write	16
	6.37 Window 1 Select	16
	6.38 Window 2 Select	16
	6.39 Window Define	16
	6.40 Window Mode	16
	6.41 Window Show	16
	6.42 Window Kill	16
	6.43 Window Flash	16
	6.44 Window Flash Speed	16
	6.45 Window Wipe Effect	17
	6.46 Window Wipe Speed	17
	6.47 Window Pattern Select	17
	6.48 Window Pattern Data	17
	6.49 Window Pattern Option	17
	6.50 Scroll text In Window	18
	6.51 Scroll Speed	18
	6.52 Select Extended Font	18
	6.53 Draw Line	19
	6.54 Auto Fade	19
	6.55 Command Delay	19
<b>7.0</b>	<b>Software Examples</b>	<b>20</b>
	7.1 Character Writes	20
	7.2 Area Commands	21
	7.3 Write Modes	21
	7.4 Graphic Writes	22
	7.5 Macros	23
	7.6 Using The Checksum	24
	7.7 Windows	24
	7.8 Flashing	25
	7.9 Scrolling Text	25
	7.10 Patterns	26
	7.11 Wipe Effects	26
<b>8.0</b>	<b>Character Sets</b>	<b>28</b>
	8.1 Mini 5x5 Font	28
	8.2 Standard 5x7 Font	28
	8.3 Cyrillic 5x7 Font	28
	8.4 Standard 7x15 Font	29
	8.5 Standard 10x14 Font	29
	8.6 Cyrillic 10x14 Font	30
<b>9.0</b>	<b>Setup</b>	<b>31</b>
	9.1 Push Button	31
<b>10.0</b>	<b>Connectors</b>	<b>32</b>
	10.1 Parallel Interface Selected – CON2	32
	10.2 Asynchronous Serial Interface Selected – CON1	32
	10.3 Synchronous Serial Interface Selected – CON1	32
	10.4 I2C Interface Selected – CON1	32
<b>11.0</b>	<b>Outline Drawing</b>	<b>33</b>
<b>12.0</b>	<b>Notation of Version</b>	<b>33</b>
<b>13.0</b>	<b>Notice For The Cautious Handling Of VFD Modules</b>	<b>34</b>

**1.0 General Description**

**1.1 Scope**

This specification covers the operating requirements of the vacuum fluorescent graphic display module GU180x32D-K610A8.

**1.2 Construction**

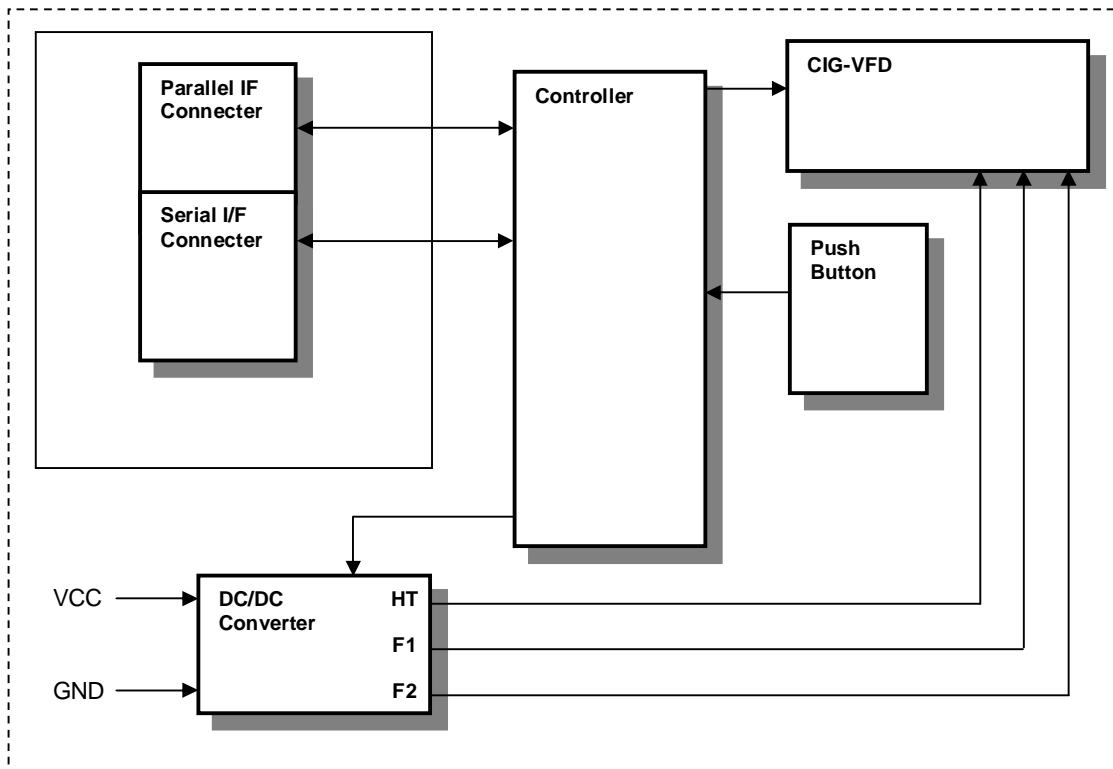
The module consists of a 180 x 32 dot graphic CIG-VFD, refresh RAM, character generator, DC/DC converter, display controller, and all necessary control logic. The interface type can be selected by a push button on the back of the module. The module can simultaneously display graphic patterns and / or characters on the screen.

**1.3 Outline**

- Power supply: + 5 V<sub>DC</sub> only
- Interface: Parallel interface (CMOS)  
C-MOS Synchronous Serial Interface  
C-MOS Asynchronous Serial Interface
- Function: Character / Graphic display  
Control commands  
Character and Graphic Image Download  
Auto key scanning  
8 User I/O Pins

Applied reliability spec: TT-99-3102

**1.4 Block Diagram**



## 2.0 Electrical Specification

### 2.1 Absolute Maximum Ratings

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition	Note
Power Supply Voltage	VCC	-0.3	-	5.5	VDC	GND=0V	5V, 0V
Logic Input Voltage	VIN	-0.3	-	V <sub>cc</sub> + 0.3	VDC	GND=0V	All Pins

### 2.2 Electrical Ratings

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition	Note
Power Supply Voltage	VCC	4.5	5.0	5.25	VDC	GND=0V	

All driving voltages for the VFD are supplied by the DC/DC converter on board.

### 2.3 Electrical Characteristics

Measuring Conditions: Ambient temperature = 25° C, VCC =5.0VDC

Parameter	Symbol	Min	Typ	Max	Unit	Condition	Note	
"H" Level Logic Input Current	I <sub>IH</sub>	-	-	1.0	uADC	VCC = 5.5V, Pin High (Absolute value)	All pins	
"L" Level Logic Input Current	I <sub>IL</sub>	-	-	1.0	uADC	VCC = 5.5V, Pin Low (Absolute value)	All pins	
Logic Input Voltage	"H"	V <sub>IH</sub>	-0.5	-	0.2 VCC	VDC	VCC = 5V	PA0-PA7, E, RXD, SCK, /SS, SIN, SCL, SDA, HB, /RES
	"L"	V <sub>IL</sub>	0.6 VCC	-	VCC + 0.5	VDC	VCC = 5V	
Logic Output Voltage	"H"	V <sub>OH</sub>	4.0	-	-	VDC	I <sub>OH</sub> = 20mA VCC = 5V	PA0-PA7, TXD, SOUT, /IRQ, SDA, MB
	"L"	V <sub>OL</sub>	0	-	0.7	VDC	I <sub>OL</sub> = -20mA VCC = 5V	
Power Supply Current 1	ICC1	360	400	440	mADC	-	All dots on	
Power Supply Current 2	ICC2	283	315	346	mADC	-	All dots off	
Power Supply Current 3	ICC3	27	30	33	mADC	-	Power off mode	

The power supply should be capable of supplying a higher current (x2) at 'power on' for 100ms

### **3.0 Optical Specifications**

Number of dots:	180 x 32(X x Y)
PCB Size:	124.0mm x 31.0mm(X x Y)
Display area:	79.05mm x 13.95mm(X x Y)
Dot size:	0.29mm x 0.29mm(X x Y)
Dot pitch:	0.44mm x 0.44mm(X x Y)
Luminance:	1000cd/m <sup>2</sup> Typ
Colour of illumination:	Blue-Green (Filter for colours)

### **4.0 Environmental Specifications**

Operating temperature:	-40 to +85 degrees C
Storage temperature:	-40 to +85 degrees C
Storage humidity:	10 to 90 % @ 25 degrees C
Vibration:	10-55-10Hz, all amplitude 1mm, 30Min., X-Y-Z (Non operating)
Shock:	392m/s <sup>2</sup> (40G) 9mS X-Y-Z, 3times each direction (Non operating)

### **5.0 Interface**

#### **5.1 Type of Interface**

The following CMOS interfaces are available on this module;  
Slow parallel interface (ENABLE, PA0 – PA7, MB, HB, /RES)  
Synchronous Serial Interface (SCK, SIN, SOUT, /SS, HB, MB, /RES)  
Asynchronous Serial Interface (RXD, TXD, HB, MB, /RES)

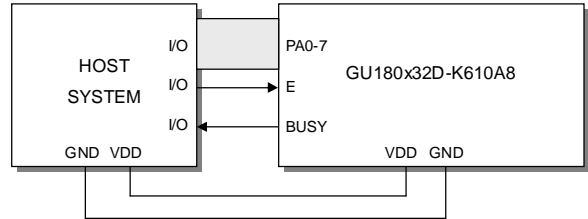
#### **5.2 Parallel Interface**

##### **5.2.1 Basic Function**

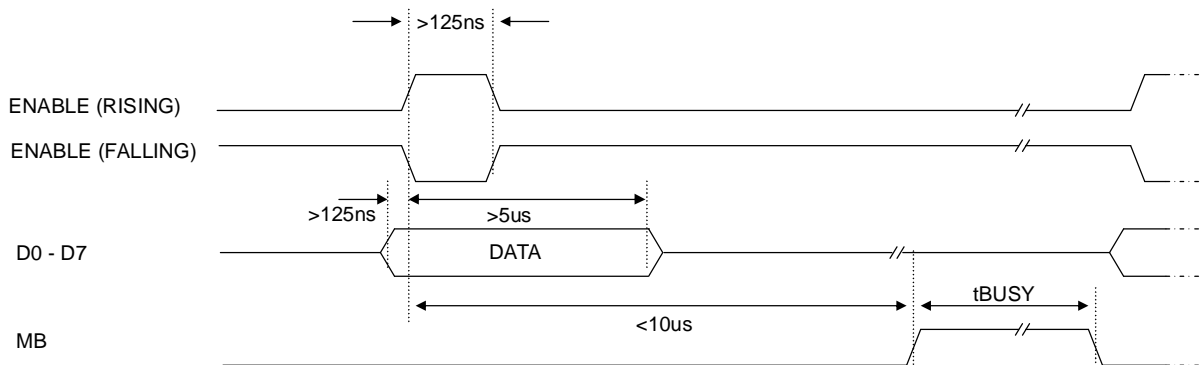
The parallel interface allows for 8 bits to be clocked into the module at a time by placing the 8 data bits on PA0 – PA7 and then clocking them into the module with the Enable line. The module sets the BUSY line upon receipt of data, and clears the line when ready to receive more data. Parallel communications are selected with a push switch at the back of the module.

### 5.2.2 Parallel Interface Timing

The 8 I/O lines can be configured as a slow parallel interface. Data on PA0-7 is clocked into the module with the Enable line this can be set to either a rising or falling edge trigger by the push switch on the back of the module. The host must keep the data stable for the time period indicated in the timing diagram. The module busy line (MB) can be used in parallel communication mode.



8-Bit Data/Command Write Timing Diagram.



The busy state can be monitored directly from CON1 pin 9. The input lines PA0-PA7 are not internally pulled up. The host system should be configured to ensure the stability of these lines.

### 5.3 Serial Interface

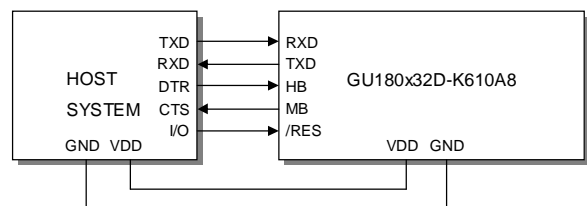
#### 5.3.1 Basic Function

Asynchronous & synchronous serial interfaces are provided at CMOS level. Serial communications are selected with a push switch at the back of the module.

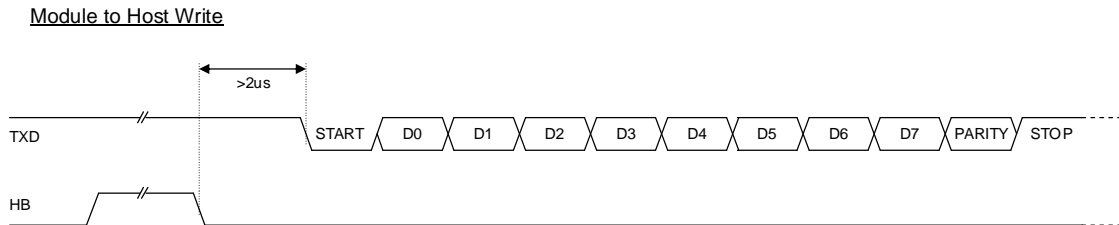
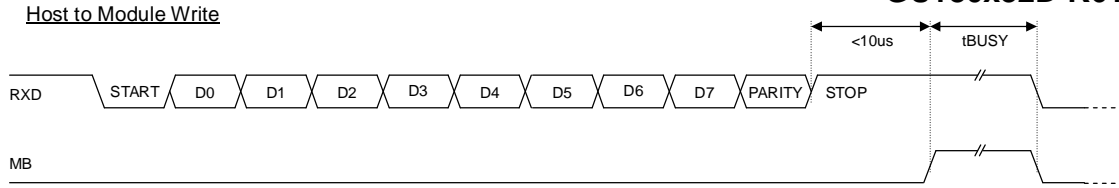
#### 5.3.2 Asynchronous Serial Interface Timing

The asynchronous communication speed and parity can be set with the push switch on the rear of the module, or with the 'SET SERIAL COMMS' command. The default settings are 19200 baud with no parity.

The host busy line (HB) stops the module from sending data to the host. The use of the HB and MB lines are optional, and can be connected together if not required.



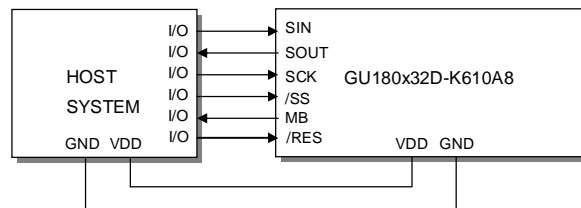
# GU180x32D-K610A8



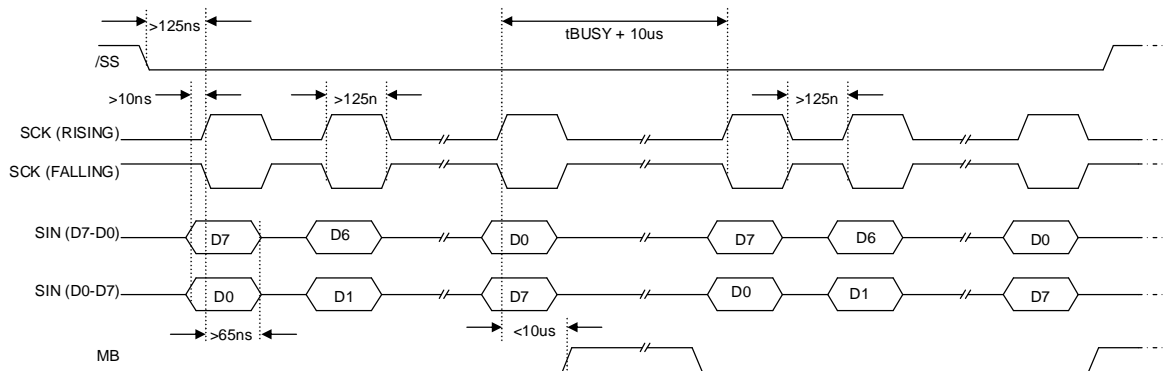
## 5.3.3 Synchronous Serial Interface Timing

With synchronous communications enabled, data can be clocked into the VFD module using the rising or falling edge of SCK. This is selectable by the push switch on the rear of the module which also sets the data order. By default, data is clocked in on the rising edge with the most significant bit sent first. The host must provide adequate delays for the module to process the data. These busy times are specified in the software command section. Alternatively the host can monitor the MB (Module Busy) line.

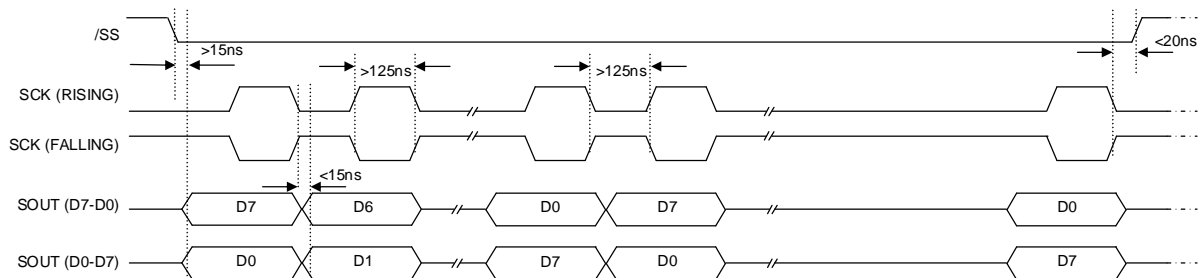
The /SS pin can be used as an enable pin if other devices are connected to the serial line, and also allows byte synchronization. The use of the /SS line is recommended, but can be permanently pulled low if not required.



### Host to Module Write



### Module to Host Write



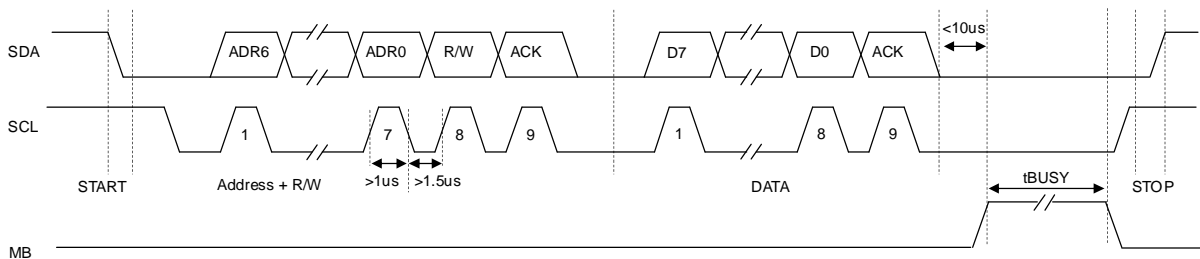
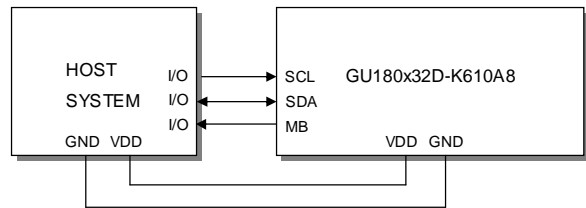
SPI Synchronous Serial Communication.



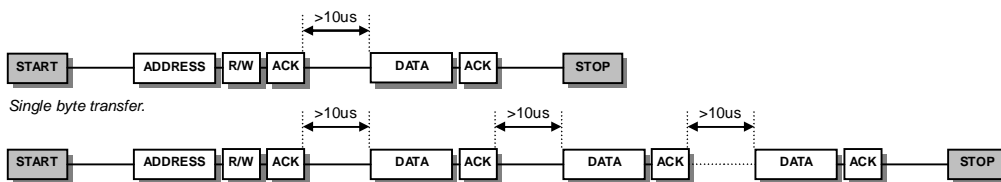
### 5.3.4 I2C Serial Interface Timing

The I2C interface operates as a slave either in 'slave receive' or 'slave transmit' mode with a fixed address of 70H. A START condition is signalled by driving SDA low while SCL is high. A STOP condition is signalled by driving SDA high while SCL is high. After a START condition is detected followed by address + R/W bit, the command / data bytes may be stored in the serial / packet buffer if selected (command data must not exceed buffer size). The module will pull SDA low during the 9<sup>th</sup> clock cycle of a data transfer to acknowledge the receipt of a byte. Additional data may be sent after an adequate delay for the module to process the data providing the host receives an Ack. If the host has not detected an Ack the data transfer must be started again by providing a STOP and START condition and address + R/W bit low. When a read command is sent the requested data is buffered, then an I2C packet must be sent with address + R/W bit high to read the command / data byte(s). The host can monitor the MB (Module Busy) line to provide adequate delays.

The SCL and SDA lines are internally pulled up with 10K resistors.



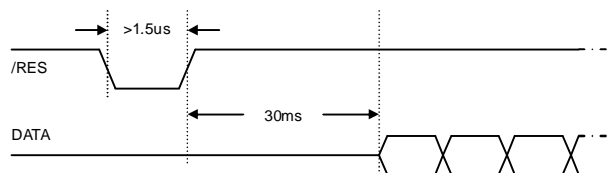
Typical I2C transmission.



Multiple byte transfer.

### 5.4 Reset

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 communication settings and set all power-up defaults. During this initialisation period, the user must delay any transmission to the module. If the user stores macros in EEPROM, the auto check and repair routine may take up to 9ms per stored byte in addition to the standard reset time.



### 5.5 Serial Buffer

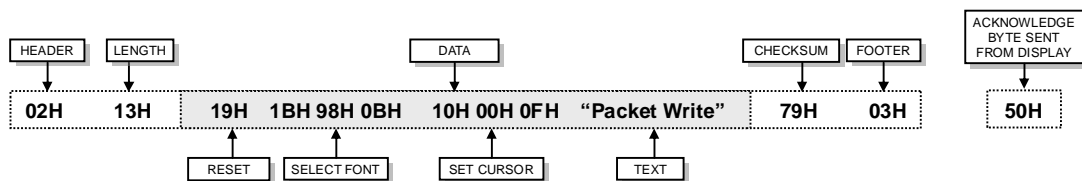
A 256-byte serial receive buffer can be activated through the setup switch on the rear of the module. This buffer can be used with any of the available serial communication modes. The buffer can also be enabled through the 'Set Serial Comms' command (see command table). Once enabled, any I/O data transmitted from the display module due to a read request or automatic I/O read, will be preceded with an identification character. Character 49H ('I') precedes I/O data bytes and 43H ('C') precedes checksum data bytes.

### 5.6 Packet Mode

The packet mode offers a more secure communication for display writing. The packet mode can be used with any of the available serial communication modes. The packet mode can also be enabled through the 'Set Serial Comms' command (see command table). Up to 32-bytes of data can be sent to the display module, encapsulated with a header (02H) and footer (03H) byte. The length of the packet should follow the header byte. An 8-bit checksum is used to validate the data. This checksum is the sum of the data bytes. The display acknowledges the packet with a 50H ('P') code for successful transfer, or a 45H ('E') for a data error.



#### Example packet transfer



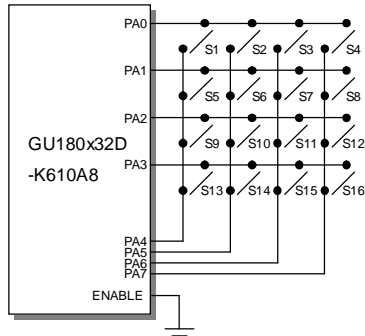
If an error occurs, the display module will discard the data, the host should then re-send the entire packet. If the packet is received correctly, then the data is placed within the 256-byte receive buffer. The receive buffer is enabled automatically when using packet mode.

### 5.7 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 an 'EEPROM Lock' command (1BH + 4CH). Once this command is issued, no further EEPROM updates can be made until it is unlocked (1BH + 55H). This feature is also accessible from the set up menu, using the push button switch on the rear of the module. During reset, EEPROM is automatically checked and repaired. When macros are stored, the module remains busy for 9ms per stored byte.

## 5.8 Keyboard Control

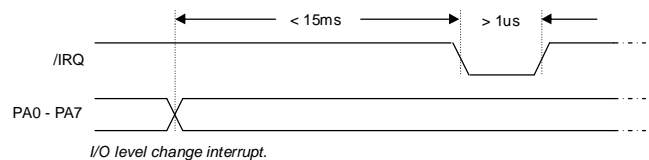
All 8 I/O lines can be configured to scan a key matrix with up to 16 keys. The 1BH + 4BH command will configure the I/O lines to key scan mode. The I/O port status will indicate the row/column position of the pressed key. The ENABLE line acts as hardware scan enable input, and should be tied to ground.



The following example enables the key-scanning mode and the automatic I/O send when using asynchronous communications.

On each key press, the I/O port status will be sent out of the asynchronous communication port. Eg, pressing key switch 1, the module will send 03H to the host system.

When using synchronous serial communications, The host can detect a level change with the /IRQ line on CON1. This allows the host to poll the port status only when needed.



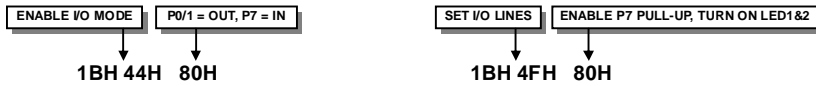
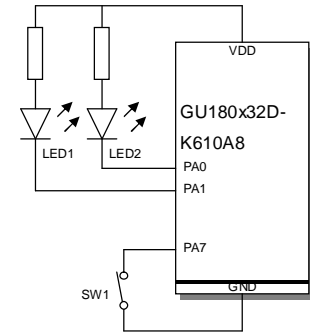
When using SPI communication the /IRQ line will indicate when a key has been pressed, the host should then issue a 'Read Port' command. The module cannot automatically send a data byte to the host so in order for the host to read the byte from the module it must also issue a 'Dummy Byte' command. After which the host will be able to read the byte received on the SPI.

When using I2C communication the /IRQ line will indicate when a key has been pressed, the host should then issue the 'Read Port' command followed by address + R. Once the module has acknowledged receipt of address + R the host can then begin to receive the port status by providing 9 clock cycles. For each clock cycle that is detected by the module a data bit will be placed on the SDA line, the host must send an acknowledgement upon receipt of the data byte by pulling SDA low during the 9<sup>th</sup> clock cycle.

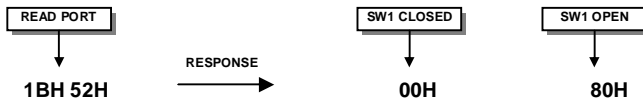
5.9 I/O Control

The module contains simple Input and Output functions for the 8 I/O lines (PA0-PA7). All inputs include an optional pull-up resistor, 30K-120K in value. The outputs can source ~5mA and sink ~30mA.

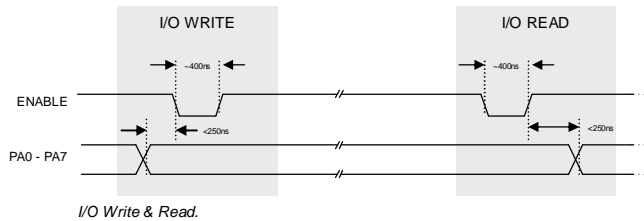
The following example sets up the I/O lines to control the 2 LED's and provide a pull-up resistor for the switch.



With asynchronous communications enabled, the status of PA0-PA7 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 I/O control is used, the Enable line can be used as an active low read or write strobe. With the I/O read command, the enable line will clock before the module reads the I/O port status. With the I/O write command, the enable line will clock after the I/O lines have been set.



## 6.0 Software Commands

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

### 6.1 Dummy Byte - Code: 00H (50μs)

Send a dummy byte which can be used to read data from the module in SPI mode.

### 6.2 Macro Start - Code: 01H – 07H (BUSY time depends on contents)

Start user defined macro 1-7

### 6.3 Back Space - Code: 08H (50μs)

Non destructive backspace. The 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.

Note, this command is disabled when using proportional font.

### 6.4 Horizontal Tab - Code: 09H (50μs)

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.

### 6.5 Line Feed - Code: 0AH (50μs)

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.

### 6.6 Home - Code: 0BH (50μs)

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.

### 6.7 Vertical Tab - Code: 0CH (50μs)

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.

### 6.8 Carriage Return - Code: 0DH (50μs)

Moves the cursor horizontal position to 00H. The vertical position is unchanged.

### 6.9 Clear EOL - Code: 0EH (2.5ms)

Clear all characters from the current cursor position to the end of the display.

### 6.10 Test - Code: 0FH (50μs)

Place module into self-test mode. The module will repetitively show a few test screens. The test mode will stop on the next received byte.

**6.11 Cursor Position - Code: 10H + X pos + Y pos** (50 $\mu$ s)

Sets the cursor position.

**6.12 Set Area - Code: 11H + X left + Y top + X right + Y bottom** (50 $\mu$ s + 1ms [Last Byte])

Fill specified area. All dots within the specified area are illuminated. Please note that the cursor position is affected with this command.

**6.13 Clear Area - Code: 12H + X left + Y top + X right + Y bottom** (50 $\mu$ s + 1ms [Last Byte])

Clear specified area. All dots within the specified area are cleared. Please note that the cursor position is affected with this command.

**6.14 Invert Area - Code: 13H + X left + Y top + X right + Y bottom** (50 $\mu$ s + 1ms [Last Byte])

Invert specified area. All dots within the specified area are inverted. Please note that the cursor position is affected with this command.

**6.15 Set Outline - Code: 14H + X left + Y top + X right + Y bottom** (50 $\mu$ s + 1ms [Last Byte])

Draw box outline. All dots within the specified outline are unchanged. Please note that the cursor position is affected with this command.

**6.16 Clear Outline - Code: 15H + X left + Y top + X right + Y bottom** (50 $\mu$ s + 1ms [Last Byte])

Clear box outline. All dots within the specified outline are unchanged. Please note that the cursor position is affected with this command.

**6.17 Set Pixel - Code: 16H** (50 $\mu$ s)

Illuminate a single pixel at the current cursor position.

**6.18 Clear Pixel - Code: 17H** (50 $\mu$ s)

Clear a single pixel at the current cursor position.

**6.19 Graphic Write - Code: 18H + Length + Data** (50 $\mu$ s + 250 $\mu$ s [Each Data Byte])

Write graphical data, length, direct to display. See write mode command (1AH) for graphic orientation and cursor movements.

**6.20 Reset - Code: 19H** (500 $\mu$ s)

Resets display to power-on defaults:

Display is cleared  
5x7 font selected  
Write Mode = 00H  
Brightness Level = 7  
VFD Power = On

**6.21 Write Mode - Code: 1AH + Data** (50µs)

Data Bit 7	Graphic Data Orientation	0 = Horizontal, 1 = Vertical (default = 0)
Data Bit 6	Cursor Movement	0 = Horizontal, 1 = Vertical (default = 0)
Data Bit 5	Cursor Direction	0 = Forward, 1 = Backwards (default = 0)
Data Bit 4	Underscore Cursor	0 = off, 1 = on (default = 0)
Data Bit 3	Underscore Cursor	0 = static, 1 = flash (default = 0)
Data Bit 2	Unused	
Data Bit 1/0	Pen Type	00 = Overwrite 01 = And 02 = OR 03 = XOR (default = 00)

**6.22 Set Macro - Code: 1BH + Macro Number + Length + Data** (50µs + 250µs [Each Data Byte])

Send macro data to EEPROM. Macro Number = 00H - 07H. Macro 0 is executed at power-up only. A maximum of 472 bytes is allowed for macro data. The display may flicker whilst writing macro data.

**6.23 Brightness - Code: 1BH + Level** (50µs)

Select one of the eight brightness levels ranging from F8H to FFH.

F8H = display off.  
F9H = minimum.  
FFH = maximum (default).

**6.24 Erase Macros - Code: 1BH + 4DH** (250ms)

Clear all downloaded macros in EEPROM. Screen may blank momentarily while macro data is being erased.

**6.25 Lock/Unlock EEPROM - Code: 1BH + 4CH + 55H** (50µs + 40ms [Last Byte])

All data contained within the non-volatile EEPROM is locked (4CH), and no changes are possible until the unlock command (55H) is executed.

**6.26 Checksum - Code: 1BH + 43H** (50µs)

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.

**6.27 Power On/Off - Code: 1BH + 50H / 46H** (50µs)

50H = Turn on VFD power supply (default).  
46H = Turn off VFD power supply (The display's contents will be preserved).

**6.28 Hex/Binary Mode - Code: 1BH + 48H / 42H** (50µs)

48H = Enable hex receive mode, character 60H is interpreted as a hexadecimal prefix.  
42H = Disable hex receive mode.

**Please note that hex mode is enabled at power up.**

**6.29 Set Serial Communications - Code: 1BH + 49H + Data**

Set Asynchronous Communications, this command takes affect at power-up or hardware reset.

Data Bit 7 = Automatic I/O Send	1 = On, 0 = Off (default = 0)
Data Bit 6 = Packet Mode	1 = On, 0 = Off (default = 0)
Data Bit 5 = Communications Buffer	1 = On, 0 = Off (default = 0)
Data Bit 4 = Unused	
Data Bit 2 = Parity	1 = Even, 0 = None (default = 0)
Data Bit 3/1/0 = baud rate	000 = 4800
	001 = 9600
	010 = 19200 (default = 0)
	011 = 38400
	100 = 57600
	101 = 76800
	110 = 1200
	111 = 2400

**6.30 Enable I/O Port - Code: 1BH + 44H + Data** (50µs + 80ms[Last Byte])

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 stored in EEPROM and will automatically be set at power up.

**6.31 Set Port Lines - Code: 1BH + 4FH + Data** (50µs)

Set Output lines on I/O port, a '1' will set 5V on the output ports, or enable the pull-ups on the inputs.

**6.32 Read Port - Code: 1BH + 52H** (50µs)

Read current I/O port status. A single byte is transmitted showing the current state of the I/O lines.

**6.33 Enable Key Scanning - Code: 1BH + 4BH** (50µs + 40ms [Last Byte])

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.

**6.34 Select Font - Code: 1CH / 1DH / 1EH** (50µs)

Select font.	1CH = proportional mini font.
	1DH = fixed spaced 5x7 font.
	1EH = fixed spaced 10x14 font.

**6.35 Graphic Area Write - Code: 1FH + X Left + Y Top + X Right + Y Bottom + Data** (50µs + 250µs [Each Data Byte])

Write graphic data within defined area. See write mode command (1AH) for graphic orientation and cursor movements.



**6.36 Character Write - Code: 20H - FFH** (500µs)

Display character from selected font.

**6.37 Window 1 Select - Code: 1BH + 80H** (50µs)

Select window 1 so that window and area command functions operate on the underlying data or text scroll.

**6.38 Window 2 Select - Code: 1BH + 81H** (50µs)

Select window 2 so that window and area command functions operate on the underlying data.

**6.39 Window Define - Code: 1BH + 82H + X Left + Y Top + X Right + Y Bottom** (50µs + 60µs [Last Byte])

Define window co-ordinates.

**6.40 Window Mode - Code: 1BH + 83H + Mode** (50µs)

Set window mode.      00H = Invert  
                                  01H = Clear  
                                  02H = Fill  
                                  03H = Pattern

**6.41 Window Show - Code: 1BH + 84H** (50µs)

Make selected window visible.

**6.42 Window Kill - Code: 1BH + 85H** (50µs)

Destroy selected window. Any scroll, flash and wipe effects will be stopped.

**6.43 Window Flash - Code: 1BH + 86H + Number** (50µs)

Flash selected window's underlying data.  
 Flash type depends on window's write mode.  
 Number = number of flashes.  
 FFH = infinite.  
 00H = stop flashing.

**6.44 Window Flash Speed - Code: 1BH + 87H + Speed** (50µs)

Set flash rate of selected window

0 = ~15ms	6 = ~250ms	12 = ~2.0sec
1 = ~30ms	7 = ~350ms	13 = ~2.5sec
2 = ~45ms	8 = ~500ms	14 = ~3.0sec
3 = ~100ms	9 = ~750ms	15 = ~3.5sec
4 = ~150ms	10 = ~1.0sec	
5 = ~200ms	11 = ~1.5sec	

Speed bits 4-7 = flash on duration, bits 0-3 = flash off duration.  
 Default speed = 88H (500ms on, 500ms off).

**6.45 Window Wipe Effect - Code: 1BH + 88H + Wipe** (50µs)

Perform a wipe action on the selected window's underlying data.

- |                             |   |
|-----------------------------|---|
| 00H = left to right cover   | 06H = top to bottom uncover             |
| 01H = right to left cover   | 07H = bottom to top uncover             |
| 02H = top to bottom cover   | 08H = horizontal centre to edge cover   |
| 03H = bottom to top cover   | 09H = horizontal edge to centre uncover |
| 04H = left to right uncover | 0AH = vertical centre to edge cover     |
| 05H = right to left uncover | 0BH = vertical edge to centre uncover   |

Note, all uncover wipes will alter the window co-ordinates.

**6.46 Window Wipe Speed - Code: 1BH + 89H + Speed** (50µs)

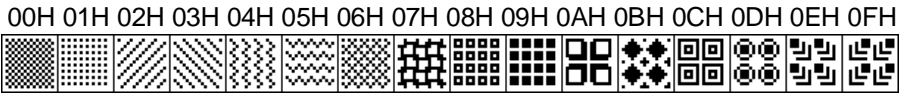
Set the wipe effect speed (pixels per second) for the selected window.

- |                       |              |
|-----------------------|--------------|
| 00H = Halt wipe       | 09H = ~157Hz |
| 01H = ~17Hz           | 10H = ~315Hz |
| 02H = ~35Hz           | 0AH = ~175Hz |
| 03H = ~52Hz           | 0BH = ~192Hz |
| 04H = ~70Hz (Default) | 0CH = ~210Hz |
| 05H = ~87Hz           | 0DH = ~227Hz |
| 06H = ~105Hz          | 0EH = ~245Hz |
| 07H = ~122Hz          | 0FH = ~262Hz |
| 08H = ~140Hz          |              |

The wipe effect duration depends upon the size of the window.

**6.47 Window Pattern Select - Code: 1BH + 8DH + Pattern** (50µs)

Select pre-defined pattern (00H-0FH) for window:



**6.48 Window Pattern Data - Code: 1BH + 8EH + Data** (50µs)

A user 16x16 pixel pattern (32 bytes) can be defined for the selected window. All data should be in vertical format with D7 uppermost.

**6.49 Window Pattern Option - Code: 1BH + 8FH + Option** (50µs)

Set window pattern options.

- |   |                                   |
|---|-----------------------------------|
| Bit 3 = invert pattern data               | 0 = Off, 1 = On (default = 0)     |
| Bit 2 = pattern alignment                 | 0 = Off, 1 = On (default = 0)     |
| Bit 1 = pattern align with edge of window | 0 = Bottom 1 = Top (default = 0)  |
| Bit 0 = pattern align with edge of window | 0 = Right, 1 = Left (default = 0) |

**6.50 Scroll Text In Window - Code: 1BH + 90H + Mode + No + Data** (50µs + no of data bytes \* 50µs [last byte])

Scroll text data within area defined by window 1.

Mode Bit 7 = Unused	
Mode Bit 6 = Unused	
Mode Bit 5 = Pad end of text with spaces	1 = On, 0 = Off
Mode Bit 4 = Scroll window's contents	1 = On, 0 = Off
Mode Bit 3 = Unused	
Mode Bit 2 = Unused	
Mode Bits 1 & 0 = Direction	00 = Scroll Up 01 = Scroll Down 10 = Scroll Left 11 = Scroll Right

No = Repeat number (00H = infinite)  
Data = Text to be scrolled with 00H signalling end of text.

Use 0DH for multi-line scrolling messages. Up to 8 rows of text can be scrolled horizontally.

**6.51 Scroll Speed - Code: 1BH + 91H + Speed** (50µs)

Set window 1 scroll speed (pixels per second)

00H = Halt Scroll  
01H = ~35Hz  
02H = ~70Hz (Default)  
03H = ~105Hz  
04H = ~140Hz  
05H = ~175Hz\*  
06H = ~210Hz\*  
07H = ~245Hz\*  
08H = ~315Hz\*

\*Horizontal scroll only.

**6.52 Select Extended Font - Code: 1BH + 98H + Font** (50µs)

Select extended font

Bits 0-2 = font number	000 = 5x5 ASCII mini font 001 = 5x7 ASCII font 010 = 10x14 ASCII font 011 = 7x15 ASCII font 100 = 5x7 Cyrillic font 101 = 10x14 Cyrillic font
Bit 3 = Proportional / fixed spacing.	1 = proportional, 0 = fixed (Default = 1)
Bits 4-6 = Horizontal font spacing 1-8 pixels.	000 = 1 pixel (Default) 001 = 2 pixels 010 = 3 pixels 011 = 4 pixels 100 = 5 pixels 101 = 6 pixels 110 = 7 pixels 111 = 8 pixels

**6.53 Draw Line - Code: 1BH + 9AH + X + Y** (50µs)

Draws line from current cursor position to specified x, y. Cursor position is updated to x, y.

**6.54 Auto Fade - Code: 1BH + 9CH + Level** (50µs)

Perform automatic fade to a defined level.

Bits 0-2 = luminance level

000 =	0%
001 =	14%
010 =	28%
011 =	43%
100 =	57%
101 =	71%
110 =	86%
111 =	100% (Default)

Bits 4-5 = speed

00 =	Fastest
01	
10	
11 =	Slowest

**6.55 Command Delay - Code: 1BH + 9FH + Delay** (50µs + delay [Last Byte])

Delay any pending commands

00H = wait for display scan to finish.

01H-F0H = multiple of 10ms delay period (10ms to 2.5 seconds).

F8H = wait for Scroll to finish.

FAH = wait for Window 1 Flash to finish.

FBH = wait for Window 2 Flash to finish.

FCH = wait for Window 1 Wipe to finish.

FDH = wait for Window 2 Wipe to finish.

Note: If scroll or flash is set to infinite repeat, the delay is ignored.

## 7.0 Software Examples

### 7.1 Character Writes

The module contains 4 font sizes, a proportional 5x5 mini-font, 5x7 pixel, 7x15, 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 FFH is treated as character data. Characters are positioned above the current cursor position, see Fig1. By default 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.

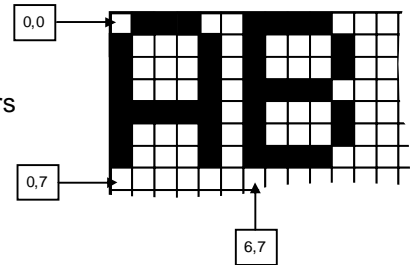
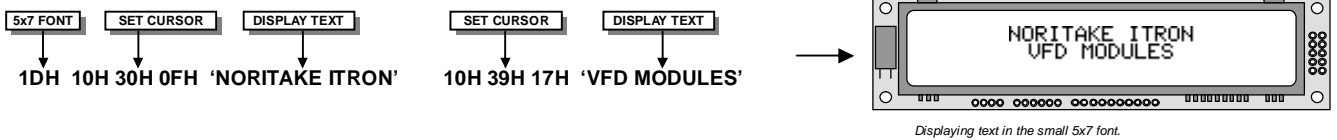
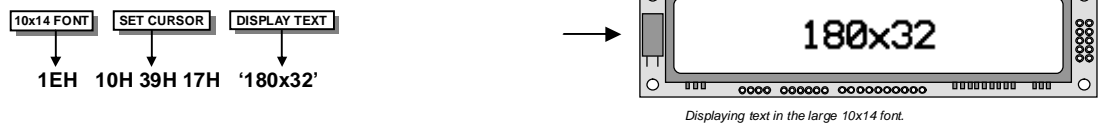


Fig 1, cursor positioning, example of writing two characters from cursor position 0,7.

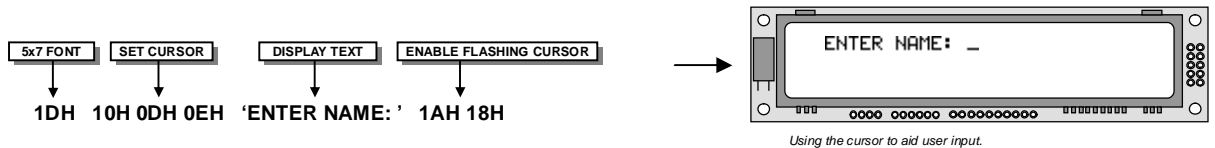
The following example displays two text messages in the centre of the display.



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

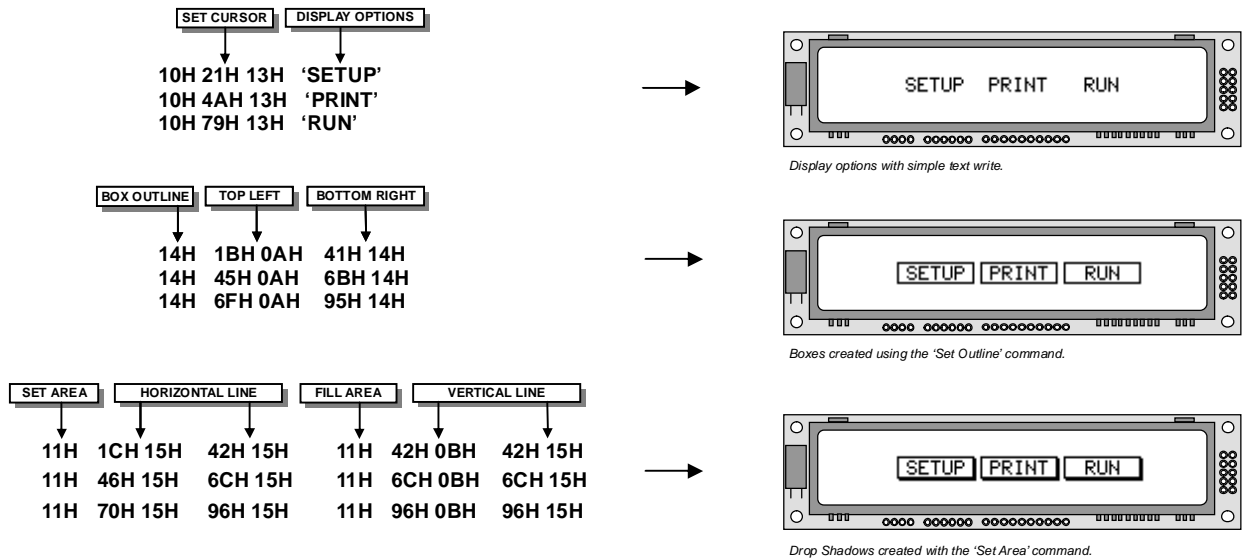


The module can display a cursor to aid character positioning and text input. The size of the cursor depends upon the currently selected font, and can be set to flash or remain static.

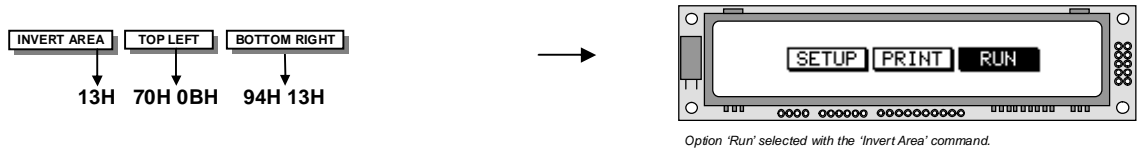


### 7.2 Area Commands

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.



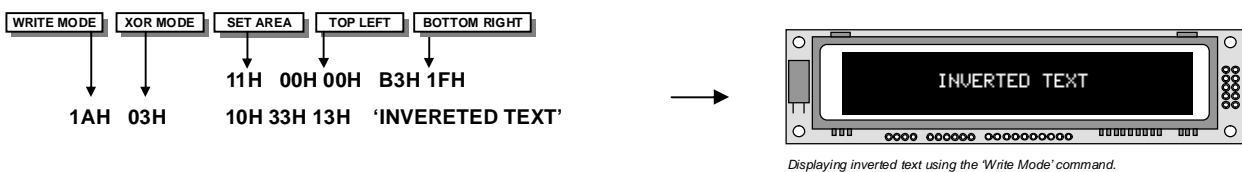
The next example uses the invert area command to select one of the options.



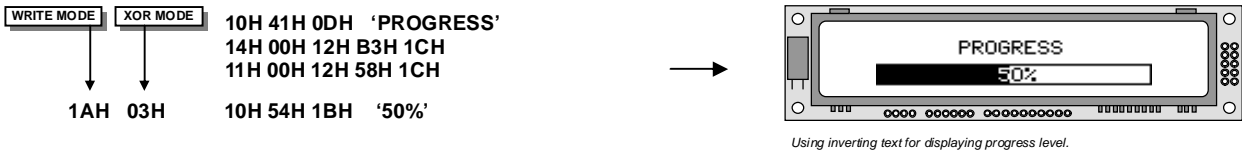
### 7.3 Write Modes

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.



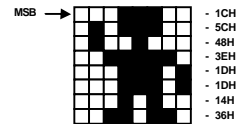
This next example uses the XOR mode to display the percentage completed on a progress bar.



### 7.4 Graphic Writes

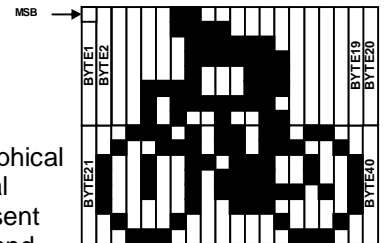
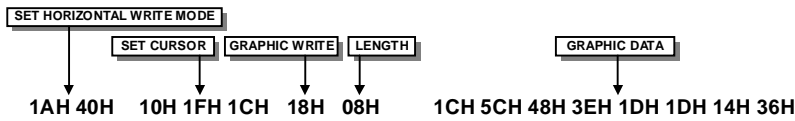
Graphical images can be displayed on the VFD module in either a horizontal or vertical byte orientation. After each graphical data write, the cursor is automatically advanced, depending upon the direction selected in the 'Write Mode' command. The most significant bit is positioned to the top (vertical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



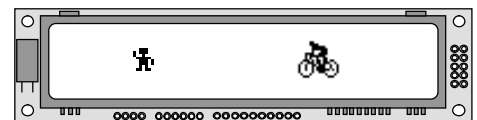
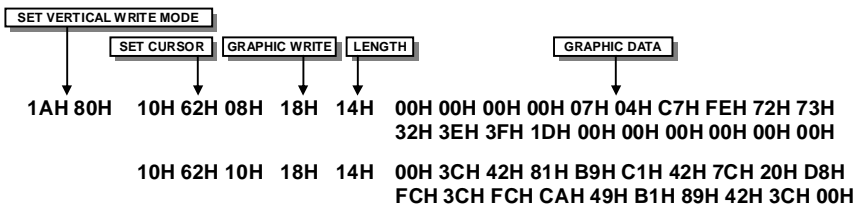
*Graphic Image using horizontal data*

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.



*Graphic Image using vertical 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.



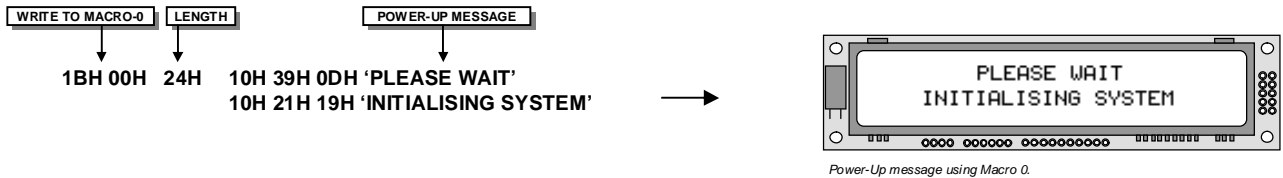
*Displaying graphic images in vertical and horizontal format.*

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.

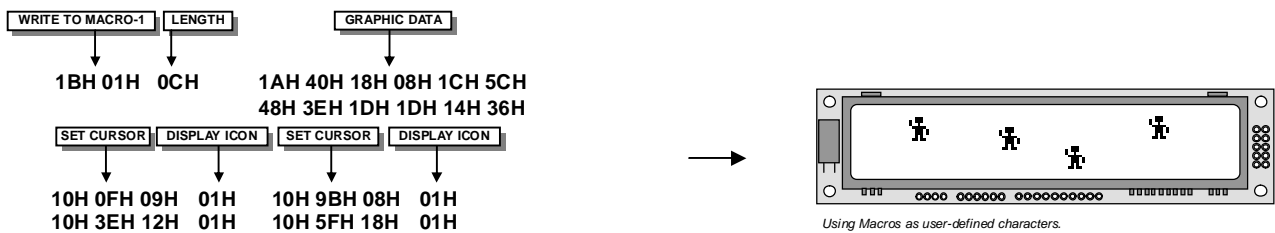
## 7.5 Macros

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.

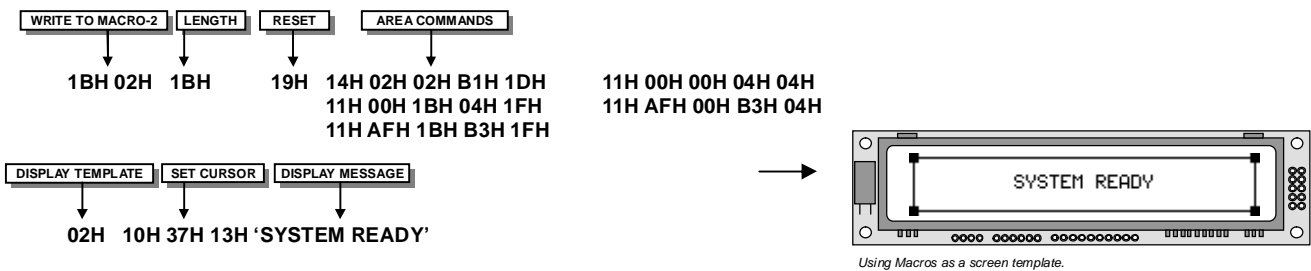
This example uses the first macro (Macro 0) to display an initial message at power-up.



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



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





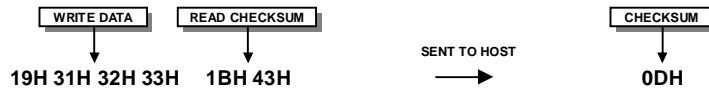
### 7.6 Using the Checksum

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.

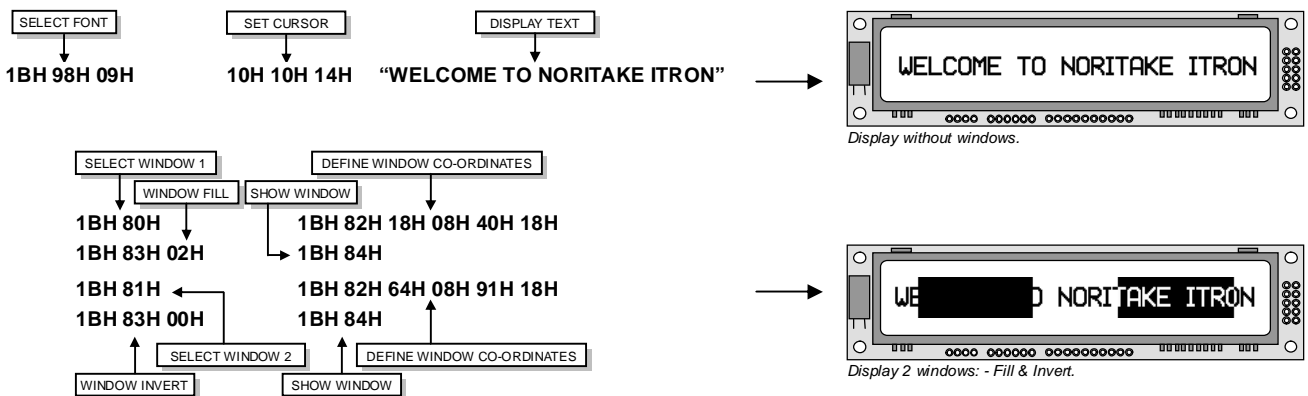


Example: Read checksum after data has been written to the display.



### 7.7 Windows

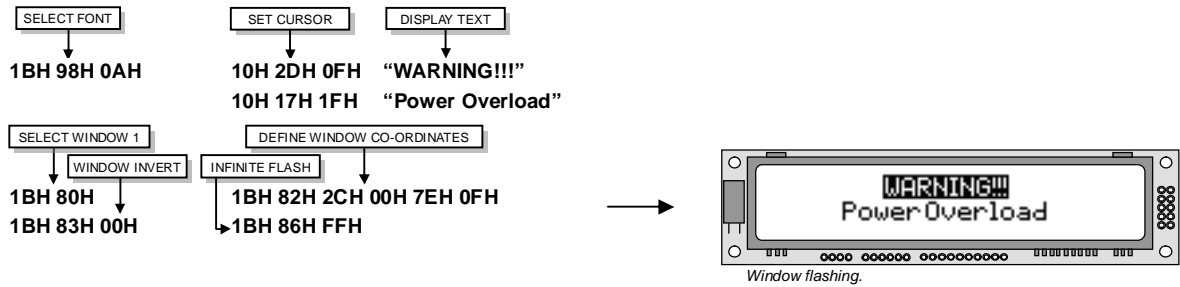
The user can create two independent windows which can manipulate underlying display information without changing the data stored in RAM. The window area can be filled with pattern data as well as flashing and wipe effects. Window 1 can be used for scrolling multi-line text.



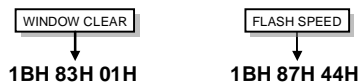
### 7.8 Flashing

The 2 windows can be utilized for display area flashing. The window flash speed and the flash amount are user-defined. The display's contents are preserved during the flash period. The flashing can also be stopped at any time.

Invert flash example.



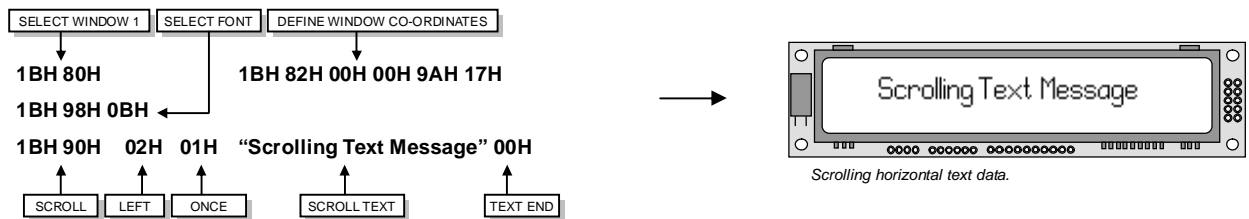
The window mode can be changed to 'clear' for a flash on/off effect. The flash on and off time duration are independent, and can be set at anytime.



### 7.9 Scrolling Text

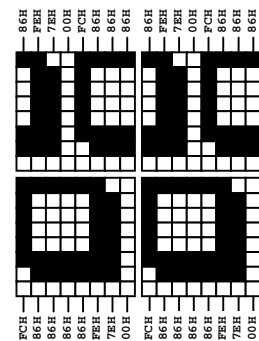
The module has the ability to scroll multi-line text within any area of the display defined by window 1. Text can be scrolled in a horizontal or vertical direction. The scroll operates independently from other commands, allowing display changes during the scroll effect. All scroll data is stored within a separate 256-byte buffer. Scrolls can be repeated a defined amount, and any font can be chosen. For multi-line scrolls, each line should be terminated by a carriage return code (0DH). A maximum of 8-lines can be scrolled horizontally.

Scroll example: -



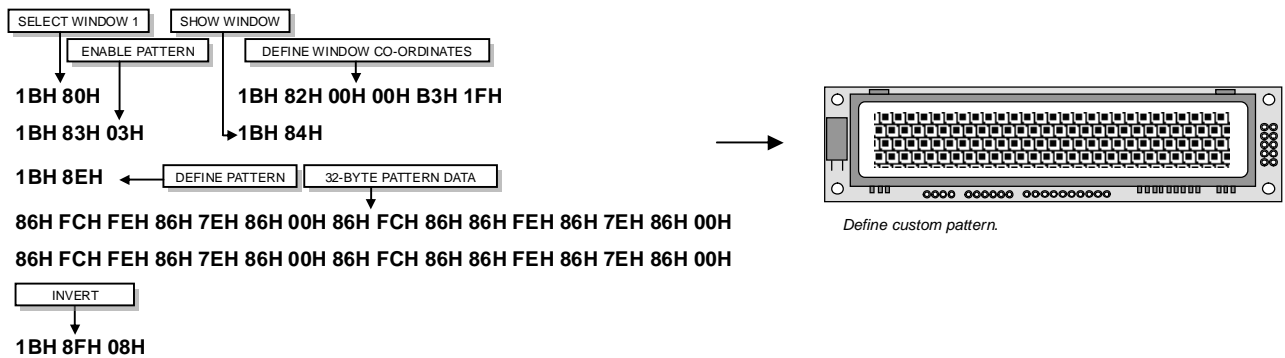
### 7.10 Patterns

One of 16 pre-defined pattern designs can be selected, and each window can have a uniquely associated pattern. Each pattern consists of 16x16 pixels and can be aligned to the top-left, top-right, bottom-left or bottom-right corner of the window. The pattern data can be inverted (negative image) if required. A user defined pattern can also be created for each window.



Custom Pattern Data.

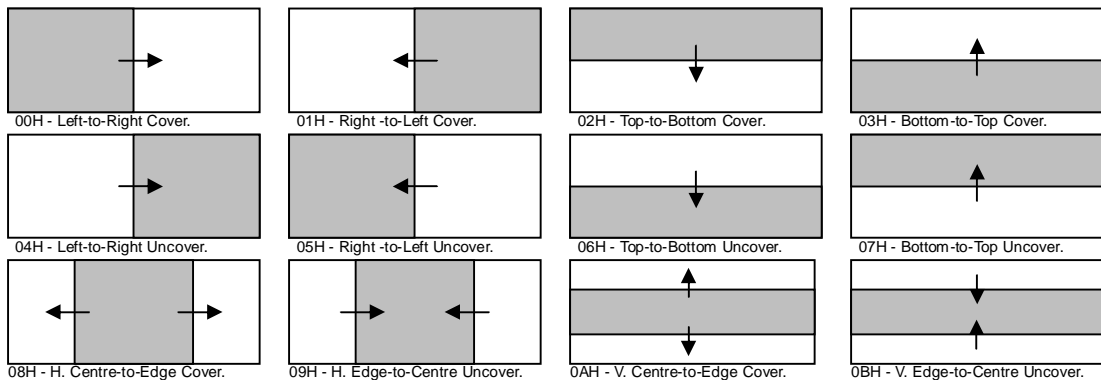
Pattern example.



### 7.11 Wipe Effects

The wipe commands modify the window co-ordinates to cover or uncover the display data. The display data is retained when using any of the wipes. The wipe effects operate independently from other commands, allowing display changes during the effect. The time duration of the wipe effect is dependant upon the defined wipe speed and the window size. The window mode is used to select the wipe fill type - Clear, Fill, Invert or Pattern. A curtain effect can be created using a pattern wipe with the align feature, the pattern data will scroll in unison with the wipe.

There are 12 wipe actions to choose from.



# GU180x32D-K610A8

Wipe examples.

SELECT WINDOW 1

1BH 80H

1BH 83H 03H

ENABLE WIPE PATTERN

CHANGE WIPE PATTERN

1BH 8DH 07H

SELECT 7x15 FONT

1BH 98H 0BH

DEFINE WINDOW CO-ORDINATES

1BH 82H 00H 00H B3H 1FH

1BH 88H 02H

START WIPE DOWN EFFECT

LOCK PATTERN TO BOTTOM OF WINDOW

1BH 8FH 06H

SET CURSOR

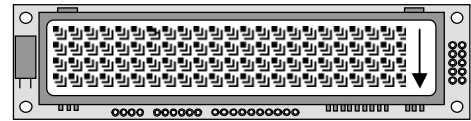
10H 12H 17H

1BH 88H 07H

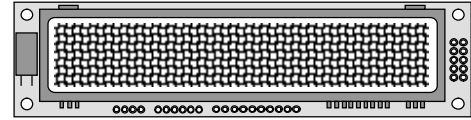
UNCOVER WIPE EFFECT

DISPLAY TEXT

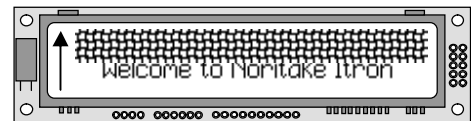
"Welcome to Noritake Itron"



Wipe down effect with pattern enabled.



Change pattern.



Uncover hidden text.

## 8.0 Character Sets

### 8.1 Mini 5x5 Font

	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	[	]	^	_	

### 8.2 Standard 5x7 Font

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
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	{	}	€	¥	
80	É	°	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯
90	Σ	Ω	≡	×	÷	°	?	∠	≥	≠	Γ	∅	∞	∞	∞	∞
A0	°	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	°
B0	°	1	2	3	½	¼	¼	¼	¼	¼	¼	¼	¼	¼	¼	¼
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

### 8.3 Cyrillic 5x7 Font

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
20																
30																
40																
50																
60																
70																
80	é	ё	ѓ	ѐ	ѕ	і	ї	ј	љ	њ	њ	ћ	ќ	-	џ	џ
90	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	
A0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
B0	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	
C0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
D0	ђ	ё	ѓ	ѐ	ѕ	і	ї	ј	љ	њ	њ	ћ	ќ	ѝ	џ	џ
E0	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
F0	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ

8.4 Standard 7x15 Font

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
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	{		}	£	¥
80	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×
90	Σ	Ω	≡	×	-	o	¿	E	≤	≥	≠	Γ	Δ	∫	∞	█
A0	°	¡	¢	£	¤	¥	¦	§	¨	©	ª	±	¸	˜	¯	—
B0	±	¹	º	»	¼	½	¾	↔	↔	↔	↔	↔	↔	↔	↔	↔
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

8.5 Standard 10x14 Font

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
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	{		}	£	¥
80	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×
90	Σ	Ω	≡	×	÷	o	¿	E	≤	≥	≠	Γ	Δ	∫	∞	█
A0	°	¡	¢	£	¤	¥	¦	§	¨	©	ª	±	¸	˜	¯	—
B0	±	¹	º	»	¼	½	¾	↔	↔	↔	↔	↔	↔	↔	↔	↔
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

8.6 Cyrillic 10x14 Font

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
20																
30																
40																
50																
60																
70																
80	Ё	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ
90	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
A0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
B0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
C0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
D0	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
E0	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ
F0	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ	ѿ

## 9.0 Setup

### 9.1 Push Button

The VFD module features a buffered asynchronous serial port and an unbuffered parallel port at CMOS level. Interface selection/set-up can be made using the single push button switch on the back of the module. Pressing the switch for the first time will display the initial configuration menu. On each subsequent switch press the menu pointer will advance. The current menu item will be selected if the switch is not pressed within 2 seconds.

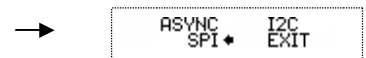
To select the required interface, press the switch until the 'COMMS' item has been selected.



Wait 2 seconds for the communication menu to be displayed. Press the switch until the required communication method is selected. The factory default interface is "SERIAL".



Wait 2 seconds for the interface menu to be displayed. Press the Switch until the required interface is selected. The factory default interface is "SPI".



Wait 2 seconds for the related communication settings and select the property to be edited.



Wait 2 seconds to display the related communication settings. The current configuration is displayed first. The factory default settings are "DATA: D7-D0", "EDGE: RISE:", "MODE: BUFFER".



*Interface selection example.*

Note: Production items can be supplied with the configuration preset and fixed.



## 10.0 Connectors

### 10.1 Parallel Interface Selected – CON2

Pin No.	Signal name	Function	Direction
1	GND	Ground	Input
2	E	Enable	Input
3	PA0	Data input	Input/output
4	PA1	Data input	Input/output
5	PA2	Data input	Input/output
6	PA3	Data input	Input/output
7	PA4	Data input	Input/output
8	PA5	Data input	Input/output
9	PA6	Data input	Input/output
10	PA7	Data input	Input/output

### 10.2 Asynchronous Serial Interface Selected – CON1

Pin No.	Signal name	Function	Direction
1	VCC	Power supply	Input
2	NC	Not Connected	
3	RXD	Receive	Input
4	NC	Not Connected	
5	GND	Ground	Input
6	NC	Not Connected	
7	TXD	Transmit	Output
8	/RES	Reset	Input
9	MB	Module Busy	Output
10	HB	Host Busy	Input

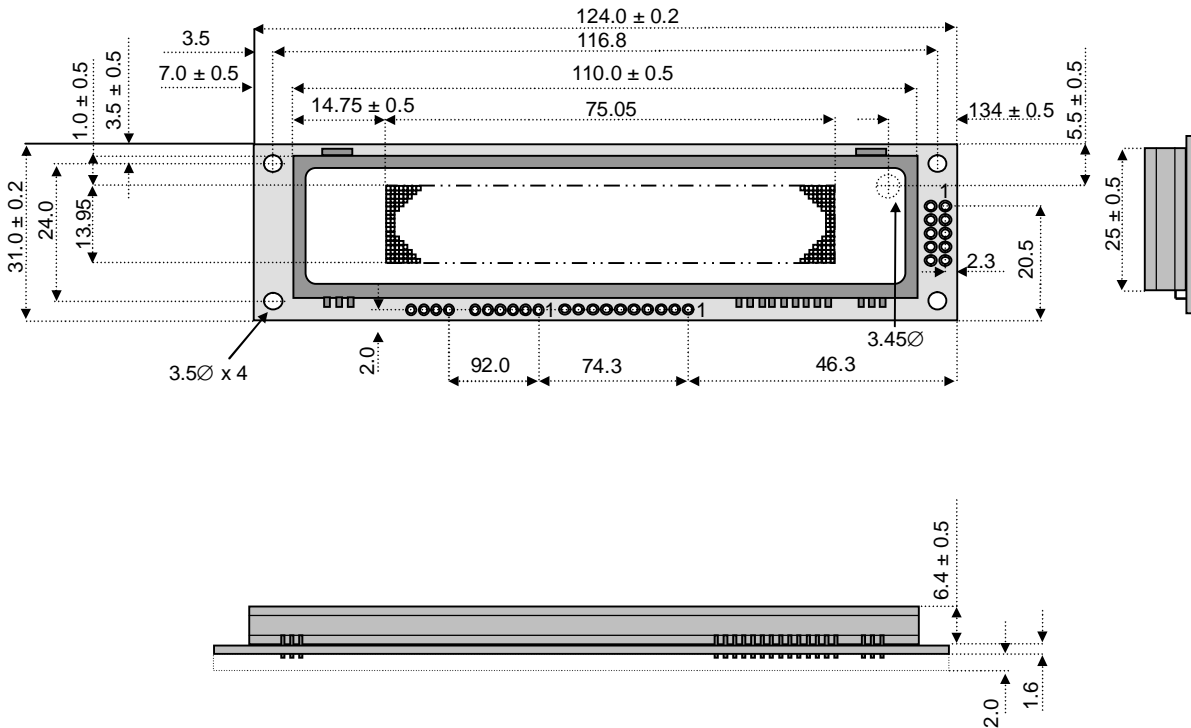
### 10.3 Synchronous Serial Interface Selected – CON1

Pin No.	Signal name	Function	Direction
1	VCC	Power supply	Input
2	SCK	Display clock	Input
3	/SS	SPI Select	Input
4	SIN	Data receive	Input
5	GND	Ground	Input
6	SOUT	Data Transmit	Output
7	/IRQ	Interrupt Request	Output
8	/RES	Reset	Input
9	MB	Module Busy	Output
10	HB	Host Busy	Input

### 10.4 I2C Interface Selected – CON1

Pin No.	Signal name	Function	Direction
1	VCC	Power supply	Input
2	NC	Not Connected	
3	SCL	Display Clock	Input
4	NC	Not Connected	
5	GND	Ground	Input
6	NC	Not Connected	
7	SDA	Data Receive/Transmit	Input/Output
8	/RES	Reset	Input
9	MB	Module Busy	Output
10	HB	Host Busy	Input

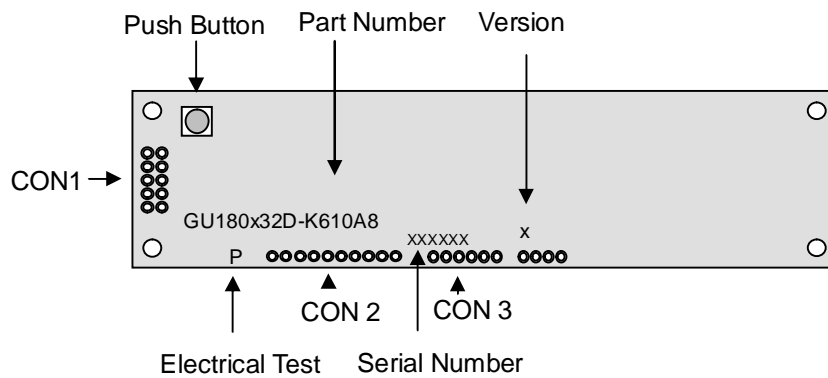
### 11.0 Outline Drawing



Dimensions in mm & subject to tolerances.

All connectors have a 2.54mm pitch

### 12.0 Notation of Version



## 13.0 Notice for the Cautious Handling of VFD Modules

### Handling and Usage Precautions:

Please carefully follow the appropriate product application notes for proper usage, safety handling, and operation standards for maximum performance.

#### [VFD tubes are made of glass]

- Because the edges of the VFD glass-envelop are not smooth, it is necessary to handle carefully to avoid injuries to your hands
- Please avoid breaking the VFD glass-envelop to prevent injury from sharp glass particles.
- The tip of the exhaust pipe is fragile so avoid shock from impact.
- It is recommended to allow sufficient open space surrounding the exhaust pipe to avoid possible damage.
- Please design the PCB for the VFD-module within 0.3 mm warping tolerance to avoid any forces that may damage the display due to PCB distortion causing a breakdown of the electrical circuit leading to VFD failure.

#### [High voltage]

- Avoid touching conductive electrical parts, because the VFD-module uses high voltage exceeding 30.100 volts.
- Even when electric power is turned off, it may take more than one minute for the electrical current to discharge.

#### [Cable connection]

- Do not unplug the power and/or data cables of VFD-modules during operating condition because unrecoverable damage may result.
- Sending input signals to the VFD-module during a power off condition sometimes causes I/O port damage.
- It is recommended to use a 30 cm or shorter signal cable to prevent functional failures.

#### [Electrostatic charge]

- VFD-modules need electrostatic free packaging and protection from electrostatic charges during handling and usage.

#### [Structure]

- During operation, VFD and VFD-modules generate heat. Please consider sufficient heat radiation dissipation using heat sink solutions.
- We prefer to use UL grade materials or components in conjunction with VFD-modules.
- Wrap and twist motion causes stress and may break VFDs & VFD modules. Please adhere to allowances within 0.3mm at the point of attachment.

#### [Power]

- Apply regulated power to the VFD-module within specified voltages to protect from failures.
- Because some VFD-modules may consume in rush current equal to twice the typical current at power-on timing, we recommend using a sufficient power capability and quick starting of the power regulator.
- VFD-module needs a specified voltage at the point of connection. Please use an adequate power cable to avoid a decrease in voltage. We also recommend inserting a power fuse for extra protection.

#### [Operating consideration]

- Illuminating phosphor will decrease in brightness during extended operation. If a fixed pattern illuminates for an extended period (several hours), the phosphor efficiency will decrease compared to the non operating phosphor causing a non uniform brightness among pixels. Please consider programming the display patterns to use all phosphor segments evenly. Scrolling may be a consideration for a period of time to refresh the phosphor condition and improve even illumination to the pixels.
- We recommend using a signal cable 30cm or less to avoid some possible disturbances to the signal.

#### [Storage and operating environment]

- Please use VFD-modules under the recommended specified environmental conditions. Salty, sulphuric and dusty environments may damage the VFD-module even during storage.

#### [Discard]

- Some VFDs contain a small amount of cadmium in the phosphor and lead in the solder. When discarding VFDs or VFD-modules please adhere to governmental related laws or regulations.

#### [Others]

- Although the VFD-module is designed to be protected from electrical noise, please plan your circuitry to exclude as much noise as possible.
- Do not reconstruct or repair the VFD-module without our authorization. We cannot assure the quality or reliability of unauthorized reconstructed VFD-modules.

### Notice:

We do not authorize the use of any patents that may be inherent in these specifications. Neither whole nor partial copying of these specifications is permitted without our approval. If necessary, please ask for assistance from our sales consultant.

This product is not designed for military, aerospace, medical or other life-critical applications. If you choose to use this product for these applications, please ask us for prior consultation or we cannot take responsibility for problems that may occur.