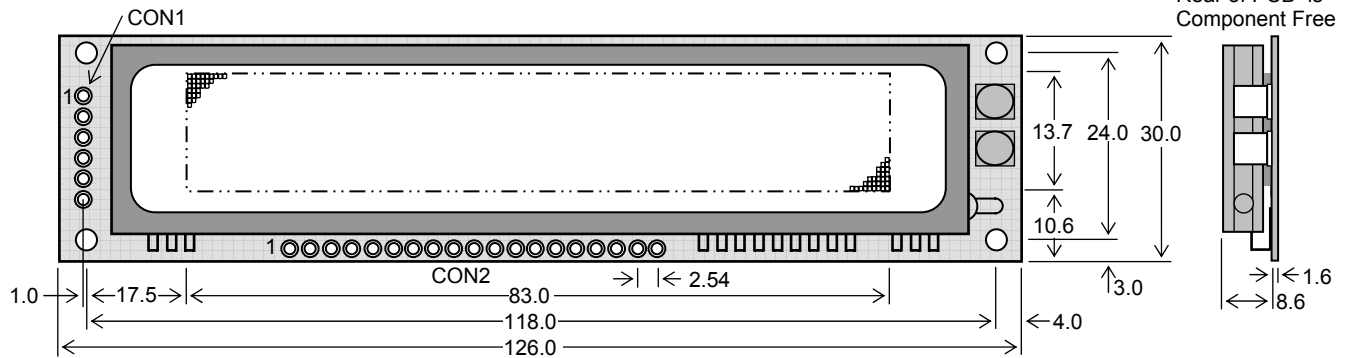


# Dot Graphic VFD Module

# 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	VDD	5.0VDC +/- 10%	GND=0V
Power Supply Current	IDD	300 mA typ.	VDD=5VDC
RS232 Serial High Input	VIH	3.0VDC min.	VDD=5VDC
RS232 Serial Low Input	VIL	0.8VDC max.	VDD=5VDC
RS232 Serial High Output	VOH	8.0VDC min.	VDD=5VDC
RS232 Serial Low Output	VOL	-4.0VDC max.	VDD=5VDC
Logic High Input	VIH	2.0VDC min.	VDD=5VDC
Logic Low Input	VIL	0.8VDC max.	VDD=5VDC
Logic High Output	VOH	2.4VDC min.	IOH=-2.0mA
Logic Low Output	VOL	0.5VDC max.	IOH=2.0mA

### CHARACTER FONT 10 x 14

	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	{		}	~	█

### OPTICAL & ENVIRONMENTAL SPECIFICATION

Parameter	Value
Display Area (X x Ymm)	83.05 x 13.69
Dot Size/Pitch (XxY mm)	0.5 x 0.6 / 0.65 x 0.77
Luminance	700 cd/m <sup>2</sup> (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.

### 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

### 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

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.

Subject to change without notice.  
IUK Doc. No. 03979 Iss.5 15Feb06



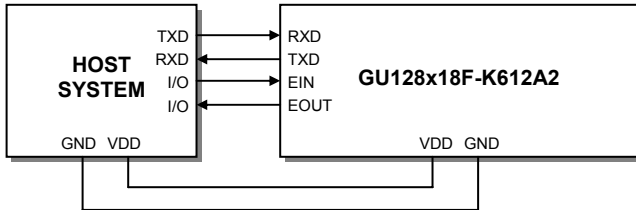
# Dot Graphic VFD Module

# GU128x18F-K612A2

## INTERFACING TO THE GU128x18F-K612A2

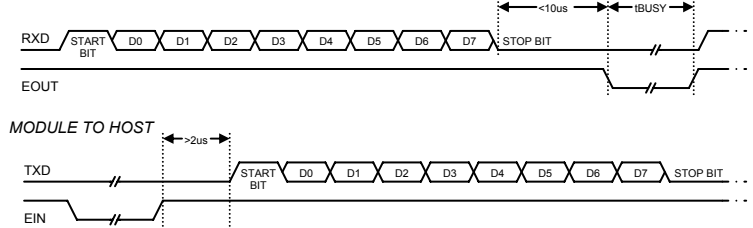
The VFD module has two serial ports for communication. The synchronous serial port uses the MOSI, MISO and SCK lines and are at TTL level. The asynchronous serial port uses lines RXD, TXD, EIN and EOUT are at RS232 levels.

### ASYNCHRONOUS SERIAL COMMUNICATION

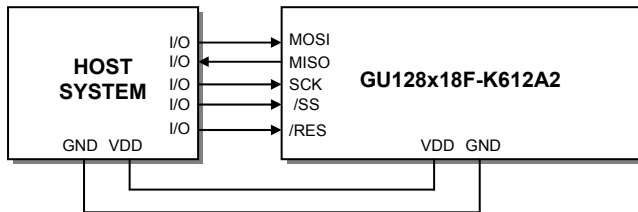


The communication speed and parity can be changed with the 'Set RS232' Software Command. These settings will be applied at power-up and with hardware reset. Linking 'EIN' and 'EOUT' will enable software handshaking. The factory default settings are 19200 baud with no parity and hardware handshaking.

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



### SYNCHRONOUS SERIAL COMMUNICATION



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.

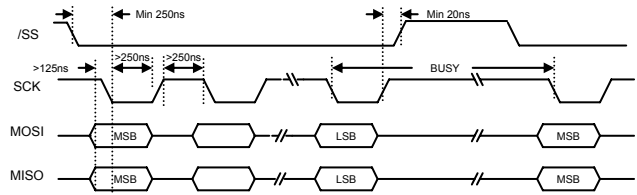
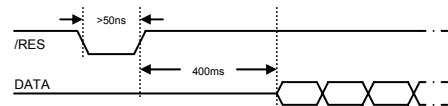


Fig2: Waveforms show TTL synchronous serial.

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

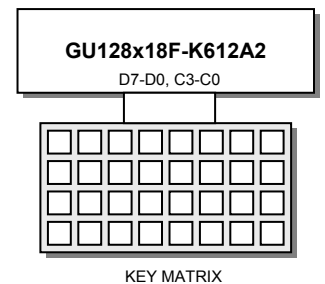


The module is reset when a low level signal is applied to the /RES line. The internal EEPROM will be checked to setup the I/O mode and the synchronous serial communications. During this initialisation period, the user must delay any transmission to the module.

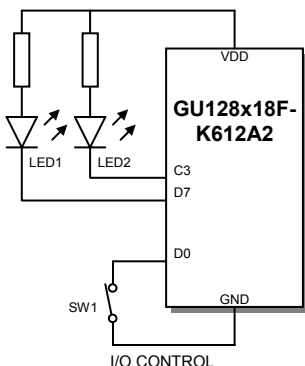
### KEY SCANNING

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: -

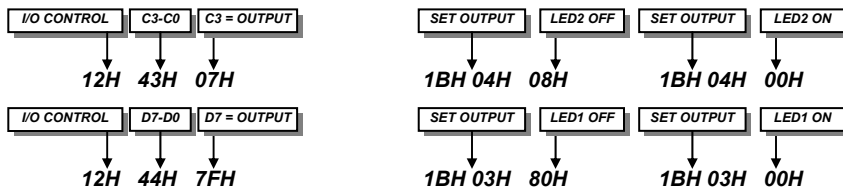


### I/O CONTROL



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.

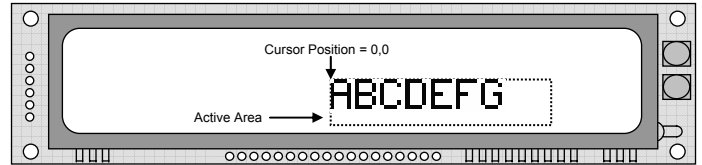
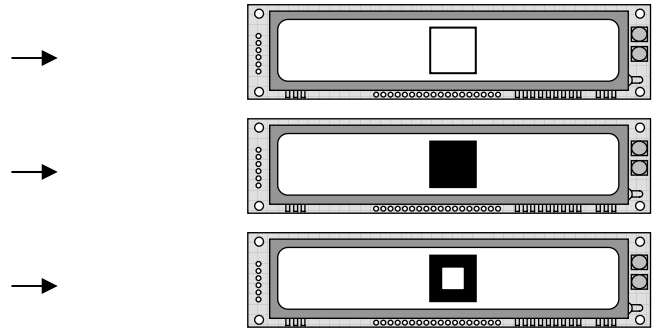
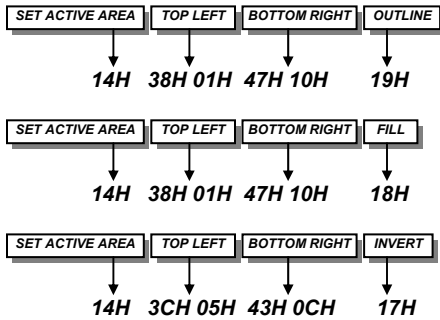


Fig2: Active Area and cursor positioning.

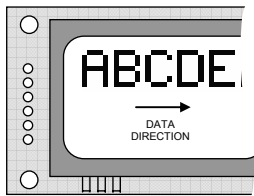
The following examples show some simple active area commands: -



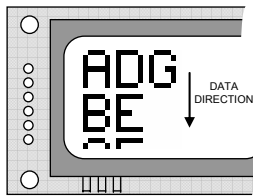
## CURSOR MOVEMENT

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.

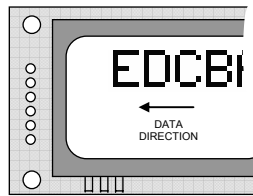
### CHARACTER WRITES



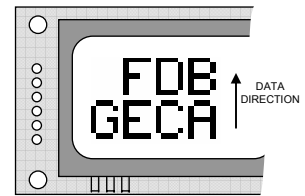
Cursor Movement = Increment  
Cursor Move = Horizontal



Cursor Movement = Increment  
Cursor Move = Vertical

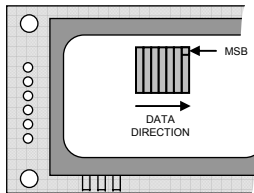


Cursor Movement = Decrement  
Cursor Move = Horizontal

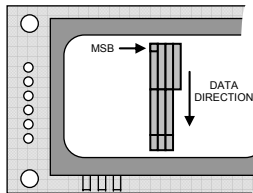


Cursor Movement = Decrement  
Cursor Move = Vertical

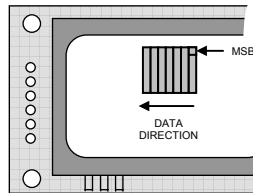
### GRAPHIC WRITES (Vertical Data)



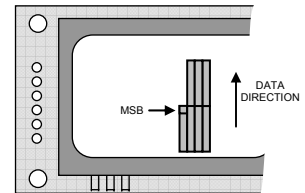
Cursor Movement = Increment  
Cursor Move = Horizontal



Cursor Movement = Increment  
Cursor Move = Vertical

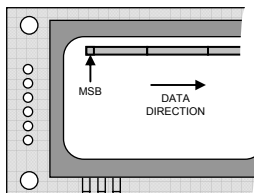


Cursor Movement = Decrement  
Cursor Move = Horizontal

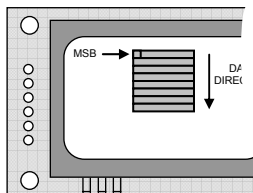


Cursor Movement = Decrement  
Cursor Move = Vertical

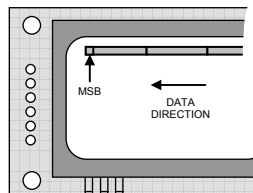
### GRAPHIC WRITES (Horizontal Data)



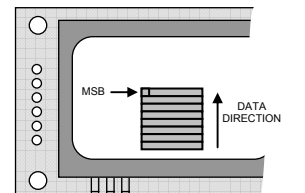
Cursor Movement = Increment  
Cursor Move = Horizontal



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.

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

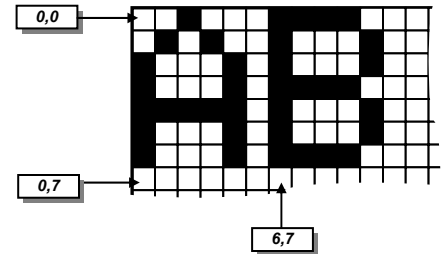
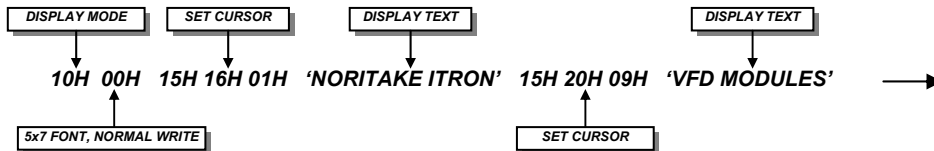
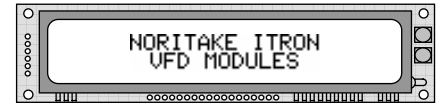
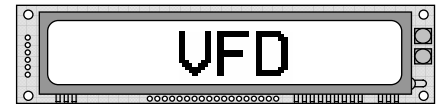
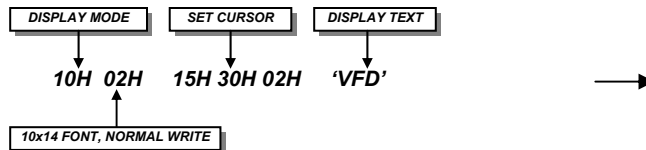


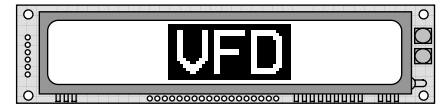
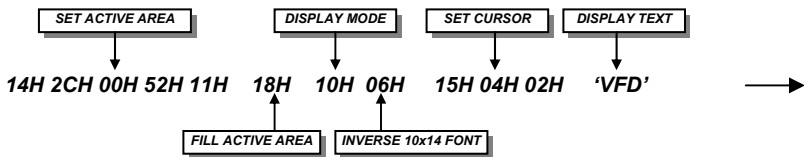
Fig1: Cursor Positioning, example of writing 2 characters from cursor position 0,0.



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.

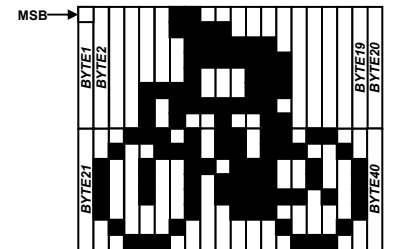
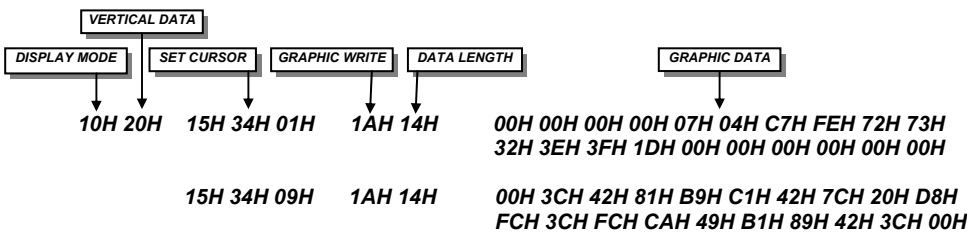
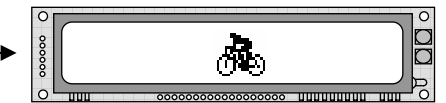


Fig3: Example Graphic Image



## 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.