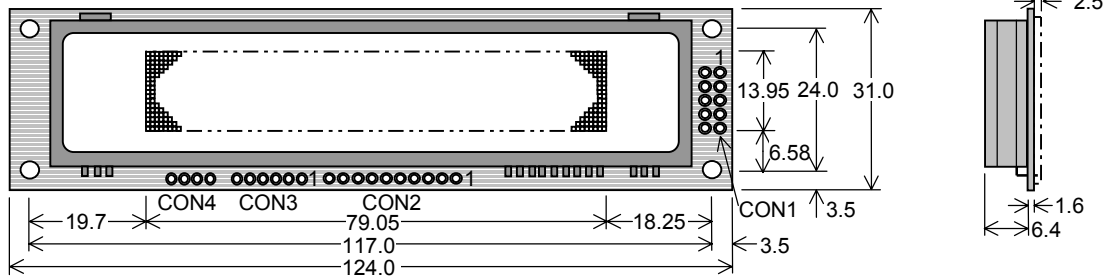


Dot Graphic VFD Module

GU180x32D-K610A4

- 180 x 32 High Brightness Dot Graphic Display
- Single 5V DC Supply
- 3 ASCII Fonts (5 x 5, 5 x 7, 10 x 14)
- Asynchronous, SPI & Parallel interfaces
- Wide Temperature -40 to +85°C

The module includes the VFD glass, VF drivers and microcontroller, character generation, interface logic and patented transformerless DC/DC converter. The interface type is selected by a pushbutton on the back of the module. Auto key scanning and general I/O is available on port PA0 – PA7. RS232/RS485 options.



Dimensions in mm & subject to tolerances.
Mounting holes 3.5mm dia.
CON3 & CON4 are not to be used in this implementation.

ELECTRICAL SPECIFICATION

Parameter	Sym	Min	Typ	Max	Unit	Condition
Supply Voltage	Vcc	4.5	5.0	5.25	V	Vss=0V
Supply Current	Icc	-	350	-	mA	Vcc=5V All dots
Logic High Input	VIH	3.0	-	Vcc	V	Vcc=5V
Logic Low Input	VIL	-0.5	-	1.5	V	Vcc=5V
Logic High Output	VOH	4.2	-	-	V	IOH = -3mA Vcc=5V
Logic Low Output	VOL	-	-	0.6	V	IOL = 20mA

ENVIRONMENTAL and OPTICAL SPECIFICATION

Parameter	Value
Display Area (XxY mm)	79.05 x 13.95
Dot Size/Pitch (XxY mm)	0.29 x 0.29/0.44 x 0.44
Luminance	1000 cd/m ² Typ
Colour of Illumination	Blue-Green (Filter for colours)
Operating Temperature	-40°C to +85°C
Storage Temperature	-40°C to +85°C
Operating Humidity (non condensing)	10 to 90% @ 25°C

SOFTWARE COMMANDS

Hex	Command	Hex	Command
01-07	Run Macro	19	Reset
08	Backspace	1A + data	Write Mode - direction
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	Enable/Disable Hex Write
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 - FF	Character Write
18 + len + data	Graphic Write		

¹ Applies to version 3 software only.

The module defaults to a 4 line of 30 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. The M(odule) Busy line indicates the module is busy when high. Connect the H(ost) Busy input to the MBusy to disable handshaking. Use the button to select the configuration which is then stored in EEPROM. To send commands as hexadecimal, prefix the 2 bytes using character 60H. Example: '10'3F'01 = Position dot x=64 y=1. To send character 60H to the display, send 60H twice.

Subject to change without notice. Doc Ref: 03872 Iss7 10 July 03

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	p
50	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	p
50	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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
10	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
20	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
30	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
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	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
90	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
00	~	a	b	c	d	e										

Dot Graphic VFD Module

GU180x32D-K610A4

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 (2.5ms)	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 stop on the next received byte.
Cursor Position (50µs)	10H + xpos + ypos	Set the cursor position.
Set Area (50µs + 1ms [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 + 1ms [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 + 1ms [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 + 1ms [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 + 1ms [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 + 250µs [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 (500µs)	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 + 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 (80ms)	1BH + 4DH	Clear all downloaded macros in EEPROM. Screen may blank momentarily while macro data is being erased.
Lock/Unlock EEPROM (50µs + 10ms [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 ASYNC comms (50µs + 10ms [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 + 20ms [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 stored 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 + 10ms [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. 1E = fixed spaced 10x14 font.
Graphic Area Write (50µs + 250µs [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. Press setup button to view.
Hex Prefix (50µs + 50µs command BUSY)	60H + dhH + dlH	Write to the display module using a 2-byte hexadecimal number. dhH = high nibble, dlH = low nibble. E.g. Sending '19' will reset the display.
Character Write (400µs)	20H - FFH	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.

NORITAKE ITRON VFD MODULES

GU180x32D-K610A4

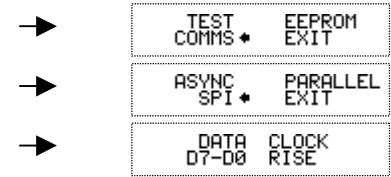
GU180x32D-K610A4 SETUP

The VFD module features two serial ports (synchronous & asynchronous) and a parallel port, all interfaces are TTL compatible. 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 interface is selected. The factory default interface is SPI.

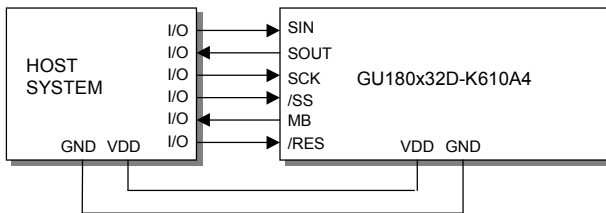
Wait another 2 seconds to display the related communication settings. The current configuration is displayed first.



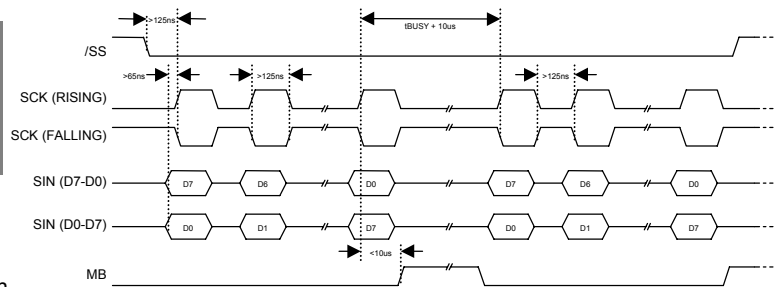
Interface selection example.

SYNCHRONOUS SERIAL COMMUNICATION

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.



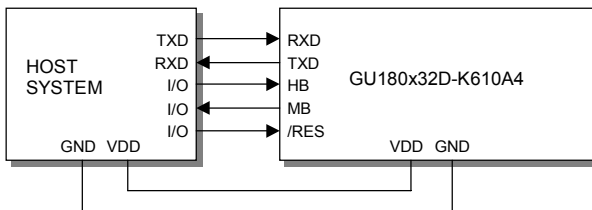
The /SS pin can be used as an enable pin if other devices are connected to the serial line, and also allows byte synchronisation. The use of the /SS line is optional, and can be permanently pulled low if required.



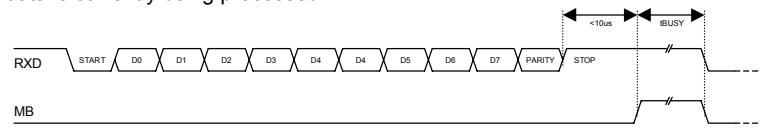
TTL Synchronous serial communication.

ASYNCHRONOUS SERIAL COMMUNICATION

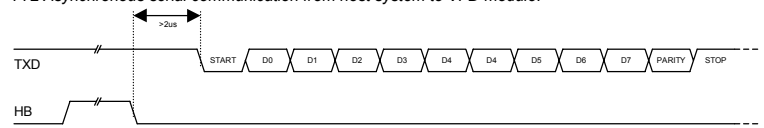
The asynchronous communication speed and parity can be set with the push switch on the rear of the module, or with the 'UART SETUP' command. The default settings are 19200 baud with no parity. Again the host must provide adequate delays for the module to process the command and data. The module busy line (MB) will go high when data is currently being processed.



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



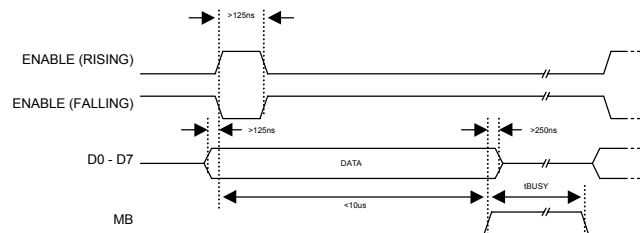
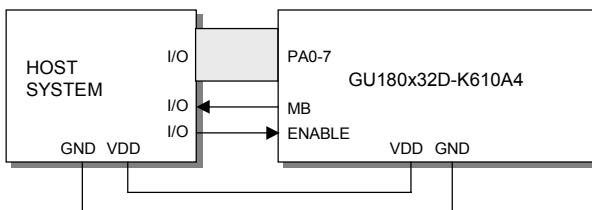
TTL Asynchronous serial communication from host system to VFD module.



TTL Asynchronous serial communication from VFD module to host system.

PARALLEL COMMUNICATION

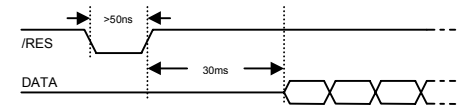
The 8 I/O lines can be configured as a slow parallel interface. Data on PA0-7 is clocked into the module with the Enable line, this can be set to either a rising or falling edge trigger by the push switch on the back of the module. The host must keep the data stable for the time period indicated in the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



Parallel Communication.

RESET TIMING

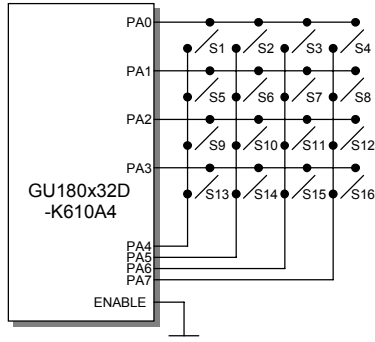
The module is reset when a low-level signal is applied to the /RES line. This will cause the module to clear the display, initialise the communication settings and set all power-up defaults. During this initialisation period, the user must delay any transmission to the module.



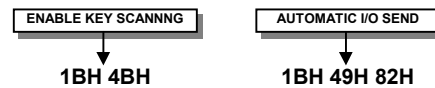
I/O level change interrupt.

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.



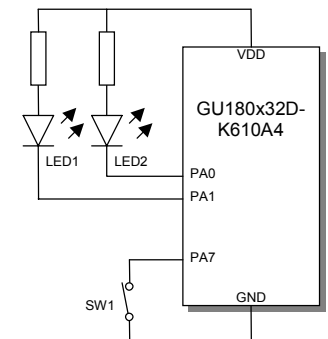
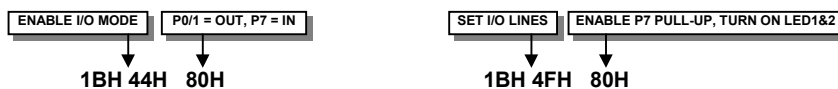
On each key press, the I/O port status will be sent out of the asynchronous communication port. e.g. pressing key switch 1, the module will send 03H to the host system.

When using synchronous serial communication, the /IRQ line will indicate when a key has been pressed, the host should then issue a 'Read Port' command to determine the I/O port status.

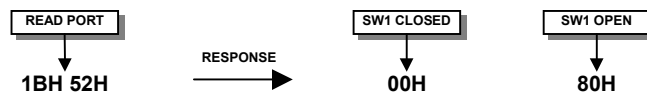
I/O CONTROL

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.

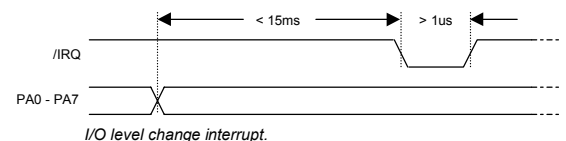
The following example sets up the I/O lines to control the 2 LED's and provide a pull-up resistor for the switch.



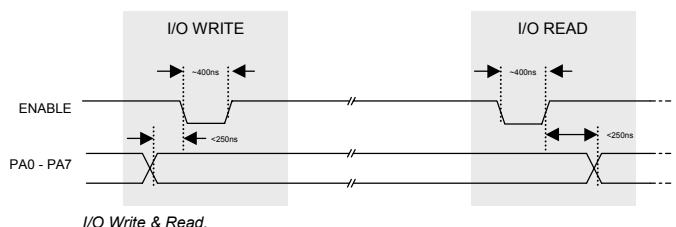
With asynchronous communications enabled, the status of PA0-PA7 can be transmitted when a change in level is detected on any pin. This automatic response mode can be enabled by using the 'UART SETUP' command. When this mode is enabled, the VFD module can reliably check port changes every 15ms. With auto send disabled (default) a manual read command is required to determine the port status.



When 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.

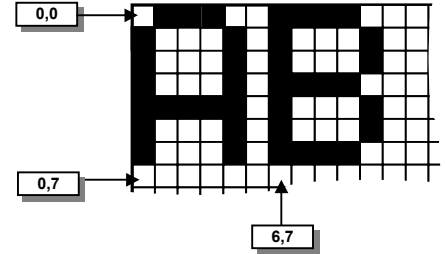


When I/O control is used, the Enable line can be used as an active low read or write strobe. With the I/O read command, the enable line will clock before the module reads the I/O port status. With the I/O write command, the enable line will clock after the I/O lines have been set.



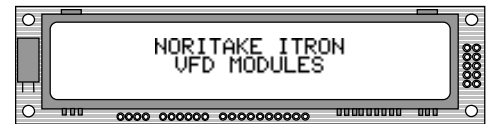
DISPLAYING TEXT

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 FFH 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.



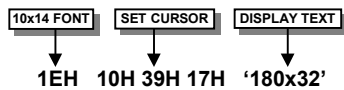
Cursor Positioning, example of writing 2 characters from cursor position 0,7.

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

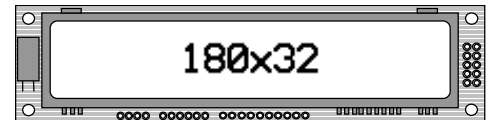
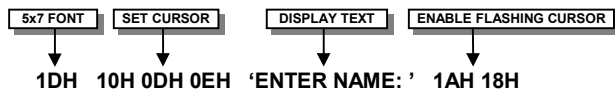


Displaying text in the small 5x7 font.

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



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.



Displaying text in the large 10x14 font.

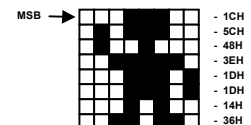
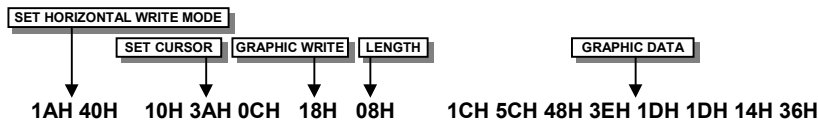


Using the cursor to aid user input.

DISPLAYING GRAPHICS

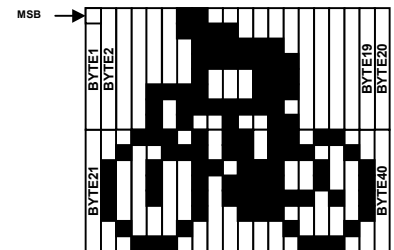
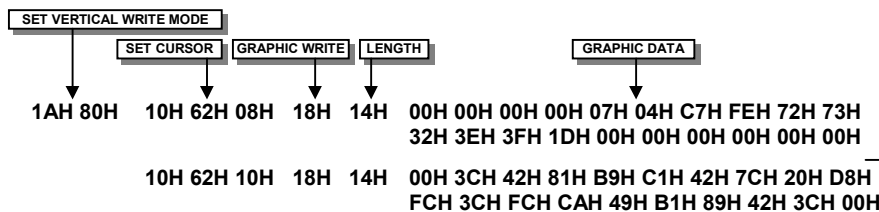
Graphical images can be displayed on the VFD module in either a horizontal or vertical byte orientation. After each graphical data write, the cursor is automatically advanced, depending upon the direction selected in the 'Write Mode' command. The most significant bit is positioned to the top (vertical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

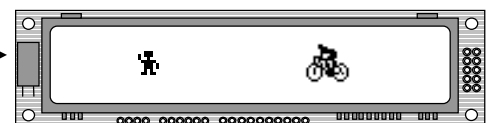


Graphic Image using horizontal data

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.



Graphic Image using vertical data



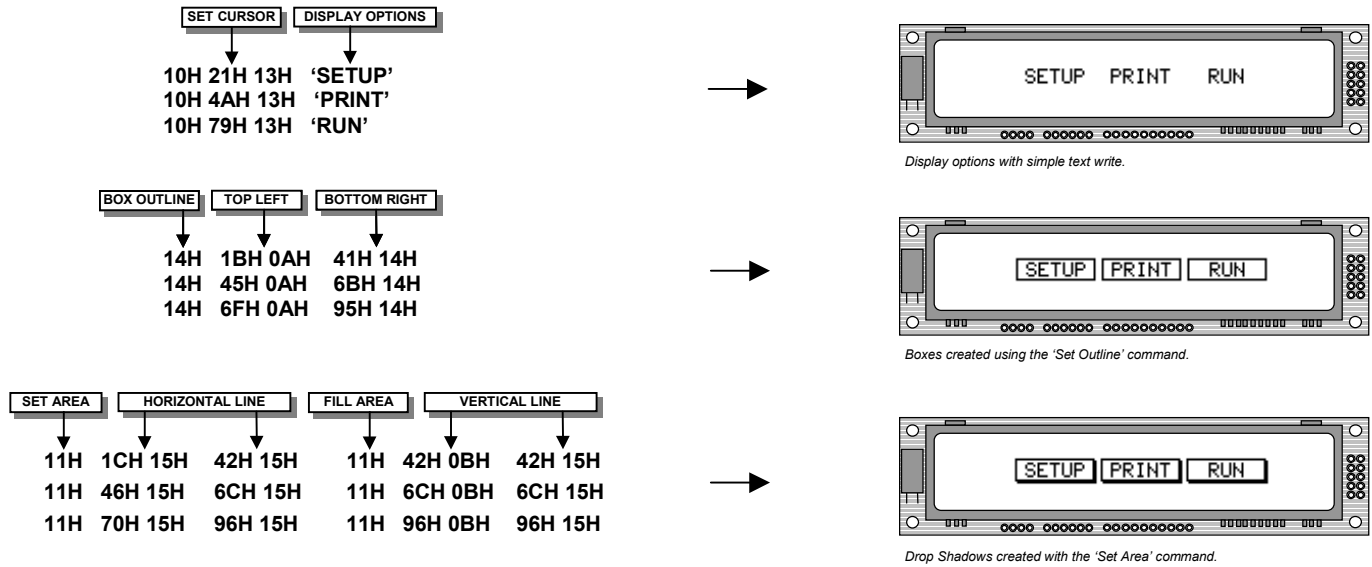
Displaying graphic images in vertical and horizontal format.

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.

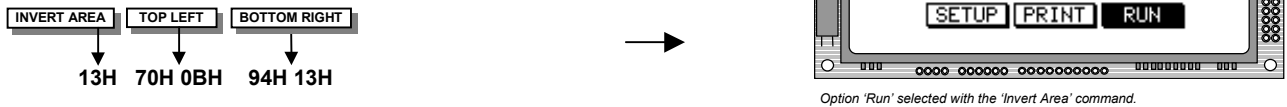
AREA COMMANDS

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.



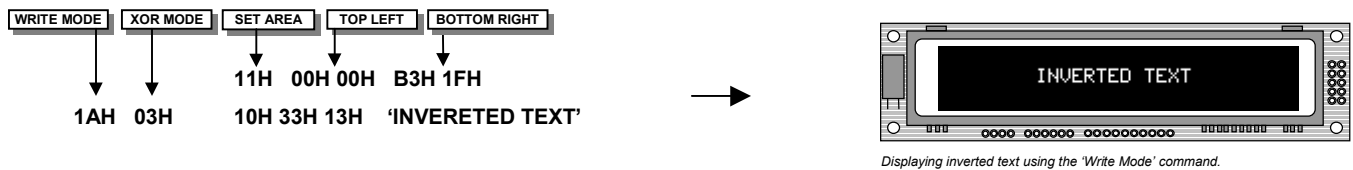
The next example uses the invert area command to select one of the options.



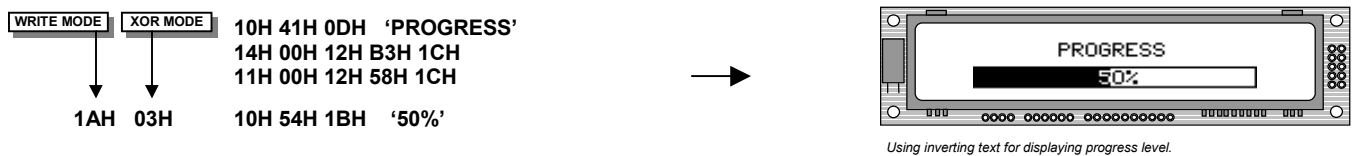
WRITE MODES

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.



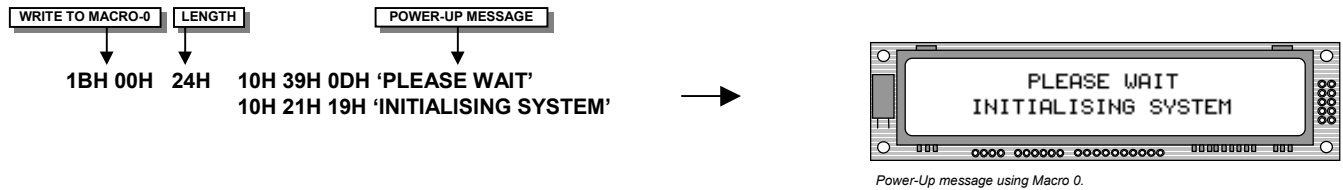
This next example uses the XOR mode to display the percentage completed on a progress bar.



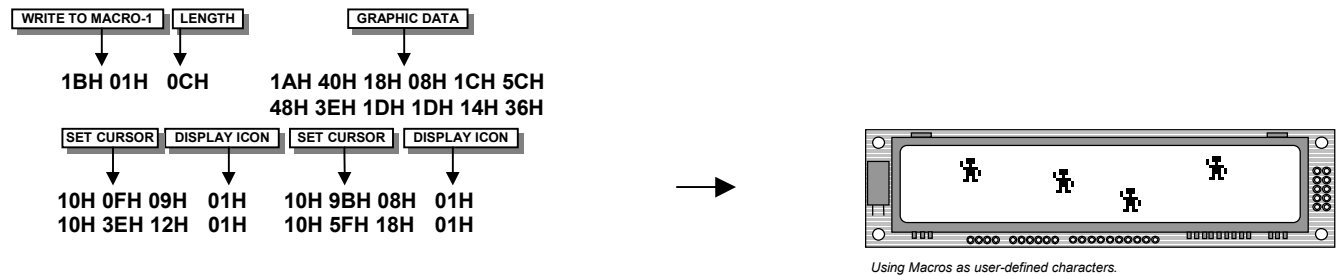
MACROS

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.

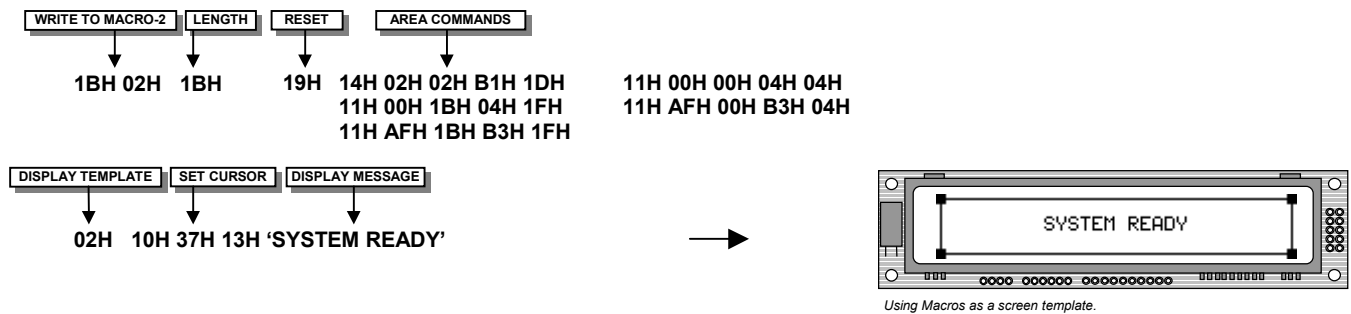
This example uses the first macro (Macro 0) to display an initial message at power-up.



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.



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). This feature is also accessible from the set up menu, using the push button switch on the rear of the module.

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.

