# Dot Graphic VFD Module GU128x64C-K610A7

- ❑ **128 x 64 High Brightness Dot Graphic Display**
- ❑ **Single 5V DC Supply**
- ❑ **5 Fonts (5x5, 5x7, 10x14, 7x15, 20x28)**
- ❑ **International & Cyrillic Fonts**
- ❑ **Asynchronous, SPI & I2C**
- ❑ **Hardware Fixed Communication Settings**

The module includes the VFD glass, VF drivers and micro-controller, character generation, interface logic and transformer-less DC/DC converter. The interface type is hardware fixed and is selected by jumper links.
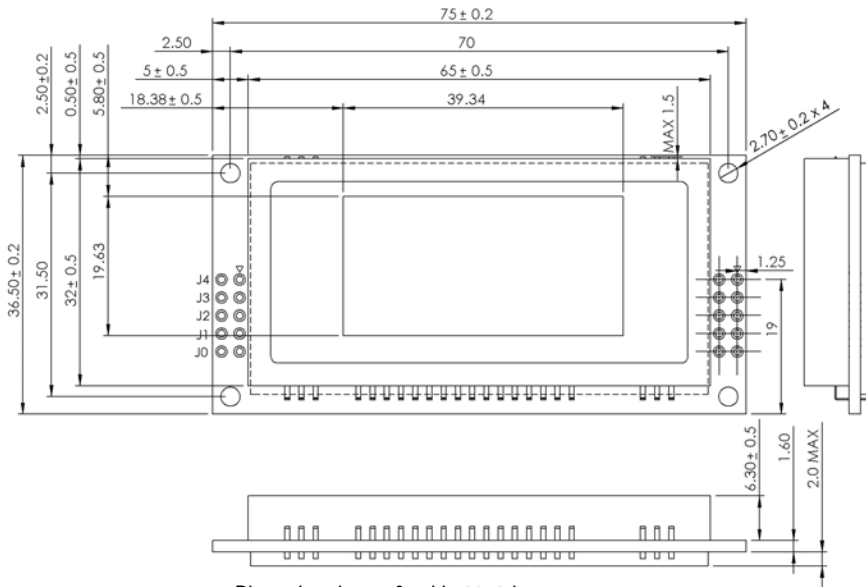
Dimensions in mm & subject to tolerances.
Uses patent applied PSU which has no inductive components.
Brown out detector active when RES is not connected.

CON2

Pin	Async	SPI	I2C
1	5V	5V	5V
2	Nc	SCK	Nc
3	RXD	/SS	SCL
4	Nc	SIN	Nc
5	0V	0V	0V
6	Nc	SOUT	Nc
7	TXD	/IRQ	SDA
8	/RES	/RES	/RES
9	MB	MB	MB
10	HB	NC	NC

Nc = Do Not Connect

CHARACTER SETS

MINI FONT

5x7, 7x15 & 10x14 FONTS

The module defaults to an 8 line of 21 character display using the 5x7 font with single pixel spacing. The cursor position auto increments after each character write. The bottom left of a character is placed at the cursor x,y. The M(odule) Busy line indicates the module is busy when high. Connect the H(ost) Busy input to the MBusy to disable handshaking. 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.
Please note that the module defaults to HEX Receive Mode.

ELECTRICAL SPECIFICATION

Parameter	Sym	Min	Typ	Max	Unit	Condition
Supply Voltage	V_{cc}	4.75	5.0	5.5	V	V_{SS}=0V
Supply Current	I_{cc}	-	290	340	mA	V_{cc}=5V All dots
Logic High Input	V_{IH}	3.7	-	V_{cc}	V	V_{SS}=0V
Logic Low Input	V_{IL}	0	-	0.3	V	V_{SS}=0V

ENVIRONMENTAL and OPTICAL SPECIFICATION

Parameter	Value
Display Area (XxY mm)	39.34 x 19.63
Dot Size/Pitch (XxY mm)	0.228 x 0.228 / 0.308 x 0.308
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 COMMAND SUMMARY

Command Type	Description
Cursor Move	ASCII commands for Back Space, Horizontal Tab, Line Feed, Home, Vertical Tab, Carriage Return and Cursor Position
Character Write	Display a character from the selected font and increments the cursor according to the write mode.
Graphic Write	Write graphical data directly to a display area in vertical orientation. It is important to disable Hex receive mode. Set or clear individual pixels.
Area Control	Set, clear, invert and outline an area of the display for easy creation of highlights and menu screens.
Window Control	Pre-define 2 areas of the display as windows in which commands and data for scroll, flash, wipe and pattern are active.
Scroll	Vertically or horizontally scroll data in either direction within an area defined by window 1. Space padding, speed and repeat options.
Flash / Wipe	Flash selected window's contents. / Perform a wipe action on the selected window with predefined user patterns / wallpaper.
Font Select	Select proportional mini font, fixed spaced 5x7 font, fixed spaced 10x14 font. Use the extended font command for 7x15 font or advanced settings.
Brightness	Set the display brightness. Level = F8H - FFH. F8H = display off. F9H = minimum, FFH = maximum *(default)*. Auto-fade to a defined level.
Power On/Off	Turn on VFD power supply *(default)*/Turn off VFD power supply (display's contents will be preserved). Current consumption is about 25mA.

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

NORITAKE ITRON VFD MODULES GU128x64C-K610A7

SOFTWARE COMMANDS

Instruction	Data Format	Description
Back Space (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 (50us)	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 (50us)	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 (50us)	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 (50us)	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 (50us)	10H + xpos + ypos	Set the cursor position.
Set Area (50us + 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 (50us + 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 (50us + 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 (50us + 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 (50us + 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 (50us)	16H	Illuminate a single pixel at the current cursor position.
Clear Pixel (50us)	17H	Clear a single pixel at the current cursor position.
Graphic Write (50us + 250us [each data byte])	18H + len + data	Write graphical data, length *len*, direct to display. Each graphic data byte is displayed vertically with MSB at the top, see write mode command (1AH) for cursor movements.
Reset (500us)	19H	Resets display to power-on defaults: - Display is cleared. 5x7 font selected. Write Mode = 00H Brightness Level = 7. VFD Power = On.
Write Mode (50us)	1AH + data	Bit 7 = graphic data orientation - 0 = horizontal, 1 = vertical *(default = horizontal)* Bit 6 = cursor movement - 0 = horizontal, 1 = vertical *(default = horizontal)* Bit 5 = cursor direction - 0 = forward, 1 = backwards *(default = forwards)* Bit 4 = underscore cursor - 0 = off, 1 = on *(default = off)* Bit 3 = underscore cursor - 0 = static, 1 = flash *(default = static)* Bit 1/0 = pen type - 00 = overwrite, 01 = AND, 02 = OR, 03 = XOR *(default = overwrite)*
Brightness (50us)	1BH + level	Set the display brightness. level = F8H - FFH. F8H = display off. F9H = minimum, FFH = maximum *(default)*.
Checksum (50us)	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 (50us)	1BH + 50H / 46H	50H = Turn on VFD power supply *(default)*. 46H = Turn off VFD power supply, display's contents will be preserved.
Hex/Binary Mode (50us)	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.**
Select Font (50us)	1CH / 1DH / 1EH	Select font. 1CH = proportional mini font. 1DH= fixed spaced 5x7 font. 1E = fixed spaced 10x14 font.
Graphic Area Write (50us + 250us [each data byte])	1FH + xl + yt + xr + yb + data	Write graphic data within defined area. Each graphic data byte is displayed vertically with MSB at the top, see write mode command (1AH) for cursor movements.
Hex Prefix (50μs + 50us + 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 (500us)	20H - FFH	Display character from selected font.
Window 1 Select (50us)	1BH + 80H	Select window 1 so that window and area command functions operate on the underlying data or text scroll.
Window 2 Select (50us)	1BH + 81H	Select window 2 so that window and area command functions operate on the underlying data.
Window Define (50us + 60us[last byte])	1BH + 82H +xl+yt+xr+yb	Define window co-ordinates.
Window Mode (50us)	1BH + 83H + mode	Set window mode: - 00H = Invert, 01H = Clear, 02H = Fill, 03H = Pattern.
Window Show (50us)	1BH + 84H	Make selected window visible.
Window Kill (50us)	1BH + 85H	Destroy selected window. Any scroll, flash and wipe effects will be stopped.
Window Flash (50us)	1BH + 86H + no	Flash selected window's underlying data. Flash type depends on window's write mode. *no* = number of flashes. FFH = infinite, 00H = stop flashing.
Window Flash Speed (50us)	1BH + 87H + speed	Set flash rate of selected window: - 0 = ~15ms 1 = ~30ms 2 = ~45ms 3 = ~100ms 4 = ~150ms 5 = ~200ms 6 = ~250ms 7 = ~350ms 8 = ~500ms 9 = ~750ms 10 = ~1.0sec 11 = ~1.5sec 11 = ~2.0sec 13 = ~2.5sec 14 = ~3.0sec 15 = ~3.5sec Speed bits 4-7 = flash on duration, bits 0-3 = flash off duration. Default speed = 88H (500ms on, 500ms off).

Command	Code	Description
Window Wipe Effect (50us)	1BH + 88H + *wipe*	Perform a wipe action on the selected window's underlying data: - 00H = left to right cover 01H = right to left cover 02H = top to bottom cover 03H = bottom to top cover 04H = left to right uncover 05H = right to left uncover 06H = top to bottom uncover 07H = bottom to top uncover 08H = horizontal centre to edge cover 09H = horizontal edge to centre uncover 0AH = vertical centre to edge cover 0BH = vertical edge to centre uncover Note: All uncover wipes will alter the window co-ordinates.
Window Wipe Speed (50us)	1BH + 89H + *speed*	Set the wipe effect speed (pixels per second) for the selected window. 00H = halt wipe 01H = ~17Hz 02H = ~35Hz 03H = ~52Hz 04H = ~70Hz 05H = ~87Hz 06H = ~105Hz 07H = ~122Hz 08H = ~140Hz 09H = ~157Hz 0AH = ~175Hz 0BH = ~192Hz 0CH = ~210Hz 0DH = ~227Hz 0EH = ~245Hz 0FH = ~262Hz 10H = ~315Hz The wipe effect duration depends upon the size of the window. Default speed = 04H (~70Hz).
Window Pattern Select (50us)	1BH + 8DH + *pat*	Select pre-defined pattern (00H-0FH) for window: -
Window Pattern Data (50us)	1BH + 8EH + data	A user 16x16 pixel pattern (32 bytes) can be defined for the selected window. All data should be in vertical format with D7 uppermost.
Window Pattern Option (50us)	1BH + 8FH + *option*	Window Pattern Options: - Bit 3 = invert pattern data. Bit 2 = pattern alignment on / off. Bit 1 = pattern align with top(1) or bottom(0) of window. Bit 0 = pattern align with left(1) or right edge of window. Default *option* = 00H (pattern alignment off & not inverted).
Scroll Text In Window 1 (50us + no of data bytes * 50us [last byte])	1BH + 90H + *mode* + *no* + *data*	Scroll text data within area defined by window 1. *mode* bits 1&0 = direction: - 00 = Scroll Up 01 = Scroll Down 10 = Scroll Left *mode* bit 4 = scroll window's contents (yes/no) *mode* bit 5 = pad end of text with spaces (yes/no) *no* = repeat number (00H = infinite) *data* = text to be scrolled with 00H = end of text. Use 0DH for multi-line scrolling messages. Up to 8 rows of text can be scrolled horizontally.
Scroll Speed (50us)	1BH + 91H + *speed*	Set window 1 scroll speed (pixels per second): - 00H = halt scroll 01H = ~35Hz 02H = ~70Hz 03H = ~105Hz 04H = ~140Hz 05H = ~175Hz* 06H = ~210Hz* 07H = ~245Hz* 08H = ~315Hz* *Horizontal scroll only. Default speed = 02H (~70Hz).
Select Extended Font (50us)	1BH + 98H + *font*	Select extended font: - bits 0-2 = font number: - 00H = 5x5 ASCII mini font. 01H = 5x7 ASCII font. 02H = 10x14 ASCII font. 03H = 7x15 ASCII font. 04H = 5x7 Cyrillic font. 05H = 10x14 Cyrillic font. 06H = 20x28 font bit 3 = proportional / fixed spacing. 1 = proportional, 0 = fixed. bits 4-6 = horizontal font spacing 1-8 pixels, where 000 = 1 pixel through to 111 = 8 pixels.
Draw Line (50us)	1BH + 9AH + *x* + *y*	Draws line from current cursor position to specified x, y. Cursor position is updated to x, y.
Auto Fade (50us)	1BH + 9CH + level	Perform automatic fade to a defined level. Bits 0-2 = luminance level, where 000 = off through to 111 = 100%. Bits 4-5 = speed, where 00 = fast through to 11 = slow.
Command Delay (50us + delay [last byte])	1BH + 9FH + *delay*	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.

Important Notes: -   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. The Back Space (08H) command is disabled when using proportional font.

GU128x64C-K610A7 SETUP

Interface selection / set-up can be made using the jumper links on CON1 as shown in the following table.

Asynchronous Settings					Synchronous Settings					Data Receive Mode	
J3	J2	J1	J0	Configuration	J3	J2	J1	J0	Configuration	J4	Configuration
O	O	O	L	9600,N,8,1	O	O	O	O	SPI, Rising, D7-D0*	O	Buffer*
O	L	O	L	19200,N,8,1	L	O	O	O	SPI, Rising, D0-D7	L	Packet
L	O	O	L	38400,N,8,1	O	L	O	O	SPI, Falling, D7-D0		
L	L	O	L	76800,N,8,1	L	L	O	O	SPI, Falling, D0-D7		
O	O	L	L	9600, E,8,1	X	O	L	O	I2C		
O	L	L	L	19200, E,8,1							
L	O	L	L	38400, E,8,1			Test Settings				
L	L	L	L	76800, E,8,1	X	L	L	O	Test Mode		

L = Linked
O = Open
X = Ignored
* Default

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.
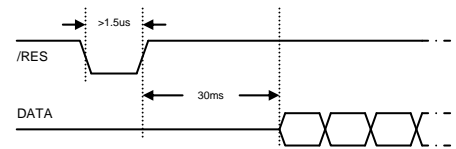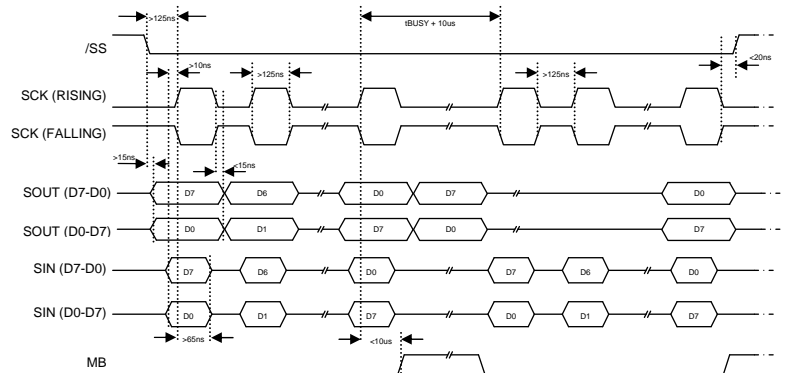

SYNCHRONOUS SERIAL COMMUNICATION (SPI)

With synchronous communications enabled, data can be clocked into the VFD module using the rising or falling edge of SCK. This is selectable by jumper links on CON1. 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.
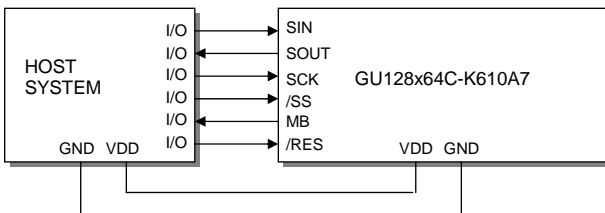

The /SS pin can be used as an enable pin if other devices are connected to the SPI bus. The use of the /SS line is optional, and can be permanently pulled low if not required. This is not recommended since /SS ensures synchronisation of the SPI bus.
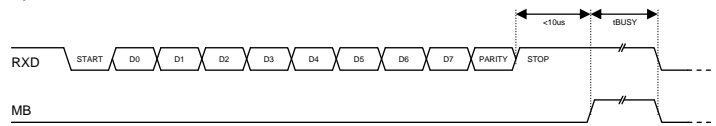
SPI Synchronous Serial Communication.
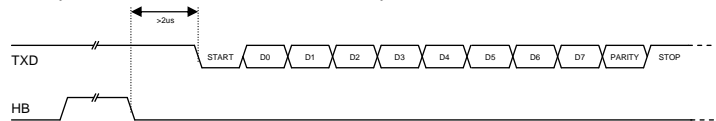
ASYNCHRONOUS SERIAL COMMUNICATION

The asynchronous communication speed and parity can be set by jumper links on CON1.

TTL Asynchronous serial communication from host system to VFD module.

TTL Asynchronous serial communication from VFD module to host system.
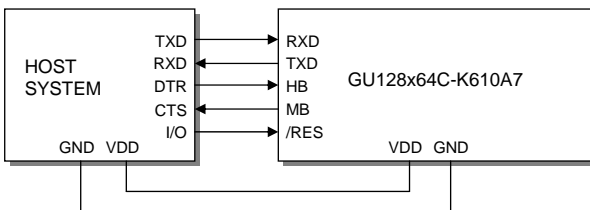
The host busy line (HB) stops the module from sending data to the host until the line falls. The use of the HB and MB lines are optional, and can be connected together if not required.
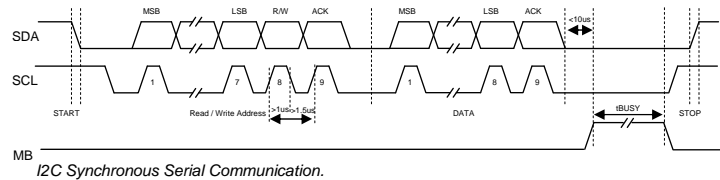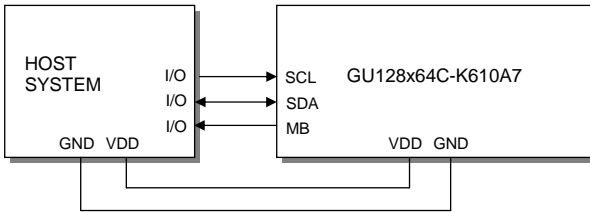
I2C COMMUNICATION

The I2C interface operates as a slave either in 'slave receive' or 'slave transmit' mode with a fixed write address of 70H and a fixed read address of 71H. A START condition is signaled by driving SDA low while SCL is high. A STOP condition is signaled by driving SDA high while SCL is high. After a START condition is detected followed by the write address, the command / data bytes are stored in the serial / packet buffer (command data must not exceed buffer size). The module will pull SDA low during the 9th 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 write address. When a read command is sent the requested data is buffered, then an I2C packet must be sent with the read address to read the command / data byte(s). The host can monitor the MB (Module Busy) line to provide adequate delays.


I2C Synchronous Serial Communication.

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

SERIAL BUFFER

The module uses a 256-byte serial receive buffer. This buffer is used with all of the available communication modes. Any data transmitted from the display module due to a read request, will be preceded with an identification character. Character 43H ('C') precedes checksum data bytes.

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 be enabled using the jumper links on CON1. 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.


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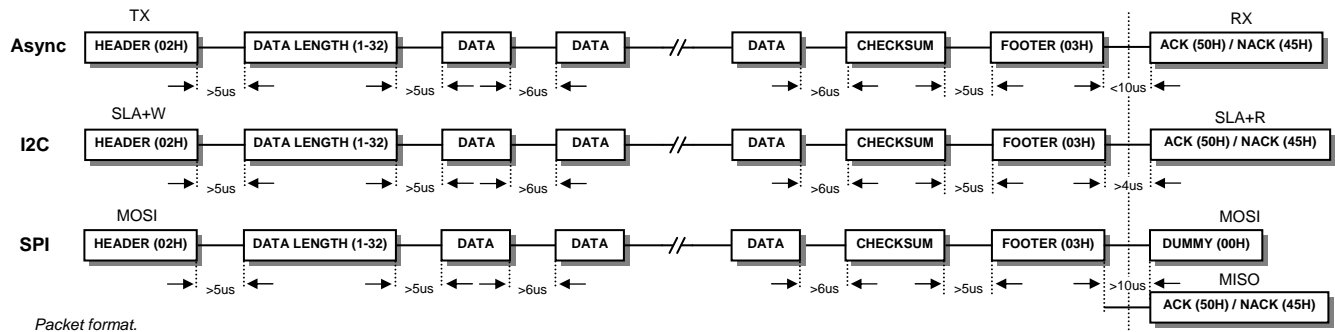
Example asynchronous packet transfer: -
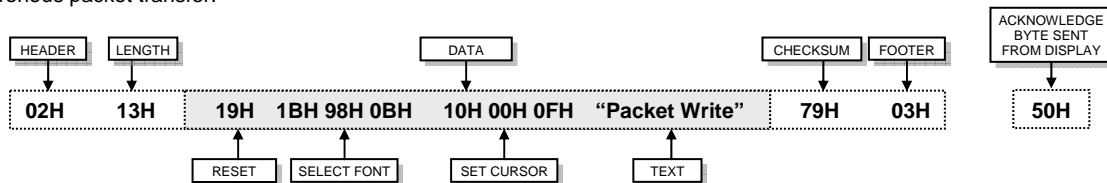
If an error occurs, the display module will discard the data, the host should then re-send the entire packet. If the packet is received correctly, then the data is placed within the 256-byte receive buffer. The receive buffer is enabled automatically when using packet mode.
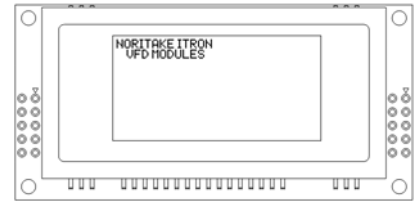
DISPLAYING TEXT

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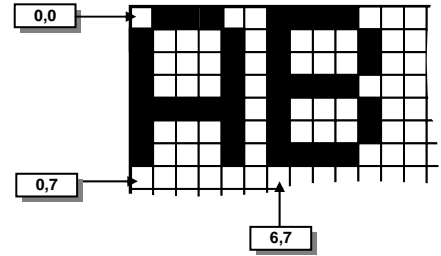
Cursor Positioning, example of writing 2 characters from cursor position 0,7.

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

5x7 FONT	SET CURSOR	DISPLAY TEXT		SET CURSOR	DISPLAY TEXT
1DH	10H 15H 1FH	'NORITAKE ITRON'		10H 1EH 27H	'VFD MODULES'

Displaying text in the small 5x7 font.

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

10x14 FONT	SET CURSOR	DISPLAY TEXT
1EH	10H 21H 27H	'128x64'
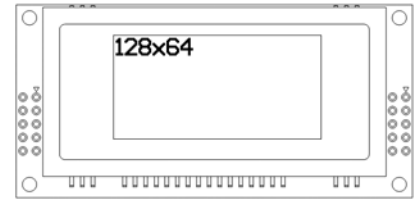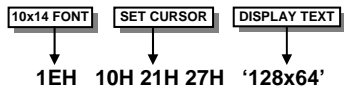
Displaying text in the large 10x14 font.

The module can display a cursor to aid character positioning and text input. The size of the cursor depends upon the currently selected font, and can be set to flash or remain static.
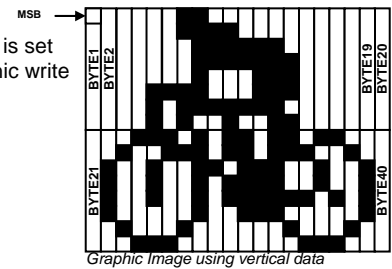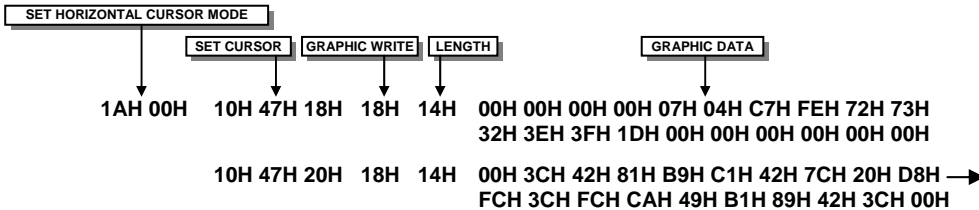
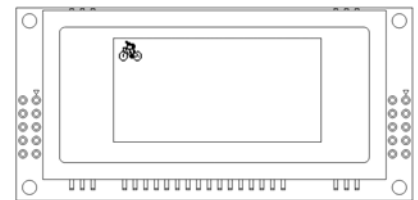| 5x7 FONT	SET CURSOR	DISPLAY TEXT	ENABLE FLASHING CURSOR
1DH	10H 0DH 0EH	'ENTER NAME: '	1AH 18H
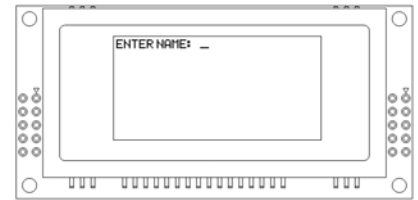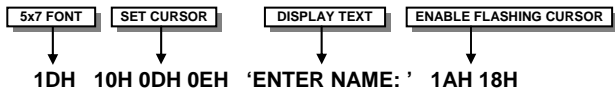
Using the cursor to aid user input.

DISPLAYING GRAPHICS

Graphical images are displayed on the VFD module in 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.

The following example displays a simple graphical image using vertical graphical data. The write mode is set to 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

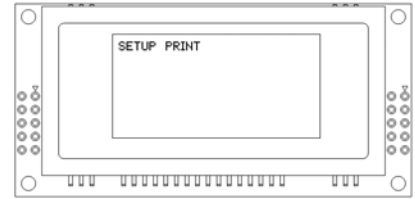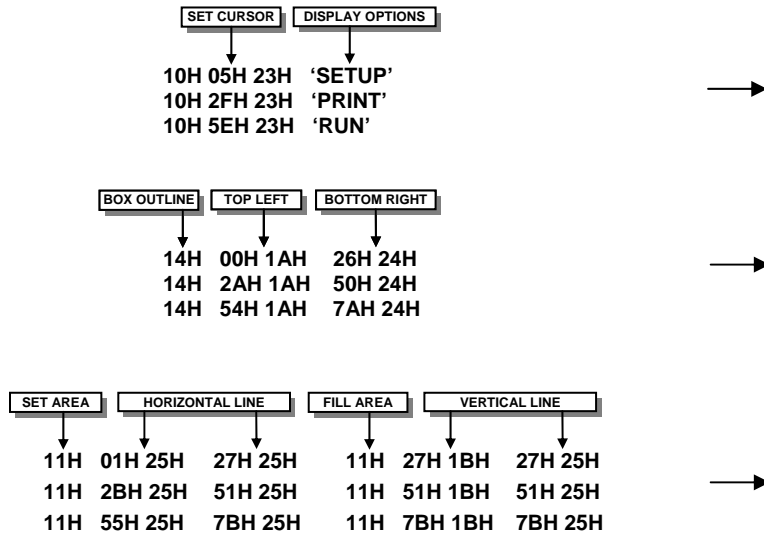| SET HORIZONTAL CURSOR MODE	SET CURSOR	GRAPHIC WRITE	LENGTH	GRAPHIC DATA
1AH 00H	10H 47H 18H	18H	14H	00H 00H 00H 00H 07H 04H C7H FEH 72H 73H 32H 3EH 3FH 1DH 00H 00H 00H 00H 00H 00H
	10H 47H 20H	18H	14H	00H 3CH 42H 81H B9H C1H 42H 7CH 20H D8H FCH 3CH FCH CAH 49H B1H 89H 42H 3CH 00H

Displaying a graphic image.

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. Each graphic data byte is displayed vertically with MSB at the top and the cursor movement is set-up using the write mode command 1AH. Unused bits are masked where the screen area is not a byte multiple.
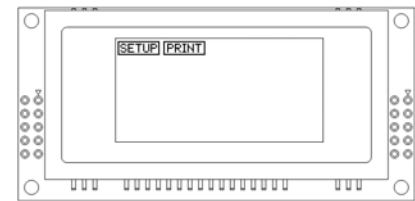
AREA COMMANDS

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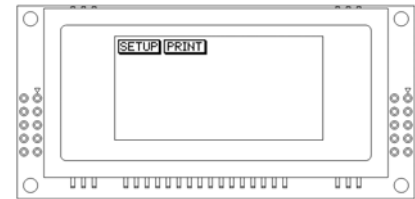
The following example displays three options for the user to select, each option is contained within a box with a shadow effect. Drawing horizontal and vertical line using the fill area command creates the shadow effect.

| SET CURSOR	DISPLAY OPTIONS

10H 05H 23H 'SETUP'
10H 2FH 23H 'PRINT'
10H 5EH 23H 'RUN'

Display options with simple text write.

BOX OUTLINE	TOP LEFT	BOTTOM RIGHT

14H 00H 1AH   26H 24H
14H 2AH 1AH   50H 24H
14H 54H 1AH   7AH 24H

Boxes created using the 'Set Outline' command.

SET AREA	HORIZONTAL LINE	FILL AREA	VERTICAL LINE

11H 01H 25H   27H 25H   11H 27H 1BH   27H 25H
11H 2BH 25H   51H 25H   11H 51H 1BH   51H 25H
11H 55H 25H   7BH 25H   11H 7BH 1BH   7BH 25H

Drop Shadows created with the 'Set Area' command.

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

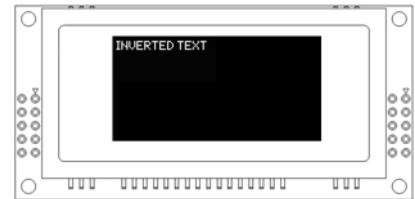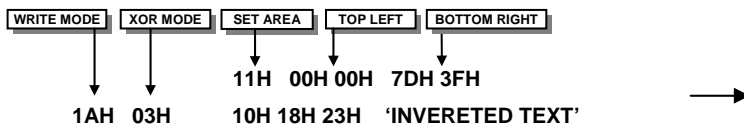| INVERT AREA	TOP LEFT	BOTTOM RIGHT

13H 55H 1BH   79H 23H
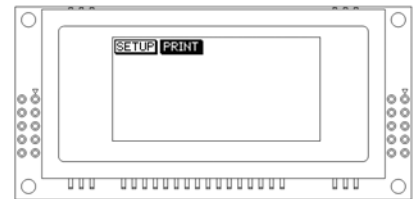
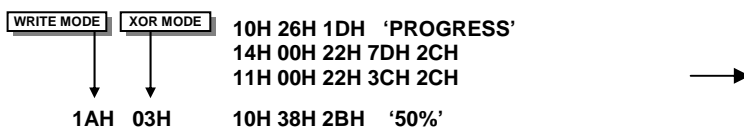*Option 'Run' selected with the 'Invert Area' command.*

WRITE MODES

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

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

WRITE MODE	XOR MODE	SET AREA	TOP LEFT	BOTTOM RIGHT

11H 00H 00H   7DH 3FH

1AH   03H   10H 18H 23H   'INVERETED TEXT'

Displaying inverted text using the 'Write Mode' command.

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

WRITE MODE	XOR MODE

10H 26H 1DH 'PROGRESS'
14H 00H 22H 7DH 2CH
11H 00H 22H 3CH 2CH

1AH   03H   10H 38H 2BH   '50%'

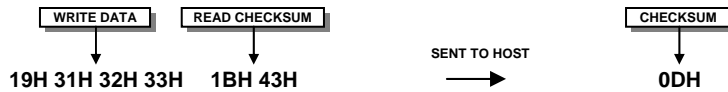*Using inverting text for displaying progress level.*

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.

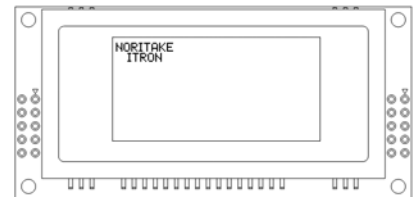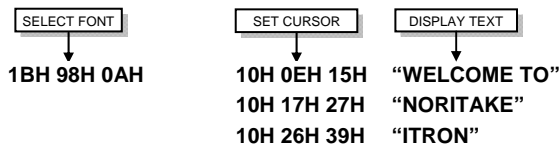Example: Read checksum at power-up, or directly after it has been cleared.

READ CHECKSUM		CHECKSUM
1BH 43H	SENT TO HOST →	**5EH**

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

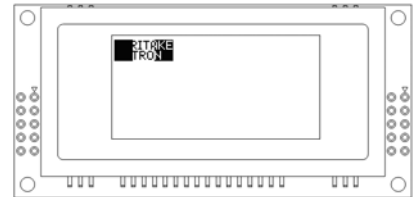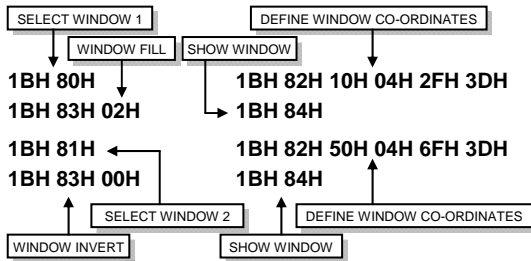| WRITE DATA	READ CHECKSUM		CHECKSUM
19H 31H 32H 33H	**1BH 43H**	SENT TO HOST →	**0DH**

WINDOWS

The user can create 2 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.
Window examples: -

SELECT FONT		SET CURSOR	DISPLAY TEXT
1BH 98H 0AH		**10H 0EH 15H**	**"WELCOME TO"**
		10H 17H 27H	**"NORITAKE"**
		10H 26H 39H	**"ITRON"**

Display without windows.

SELECT WINDOW 1   DEFINE WINDOW CO-ORDINATES
WINDOW FILL   SHOW WINDOW

1BH 80H
1BH 83H 02H   **1BH 82H 10H 04H 2FH 3DH**
    1BH 84H

1BH 81H ←   **1BH 82H 50H 04H 6FH 3DH**
1BH 83H 00H   **1BH 84H**

SELECT WINDOW 2   DEFINE WINDOW CO-ORDINATES
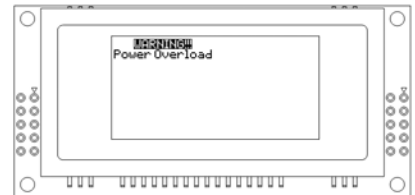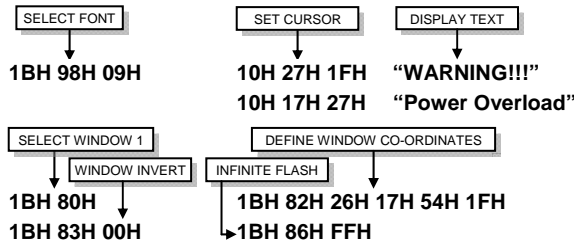WINDOW INVERT   SHOW WINDOW

Display 2 windows: - Fill & Invert.

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: -

SELECT FONT		SET CURSOR	DISPLAY TEXT
1BH 98H 09H		**10H 27H 1FH**	**"WARNING!!!"**
		10H 17H 27H	**"Power Overload"**

SELECT WINDOW 1   DEFINE WINDOW CO-ORDINATES
WINDOW INVERT   INFINITE FLASH

1BH 80H
1BH 83H 00H   **1BH 82H 26H 17H 54H 1FH**
    1BH 86H FFH

Window flashing.

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: -

WINDOW CLEAR	FLASH SPEED
1BH 83H 01H	**1BH 87H 44H**
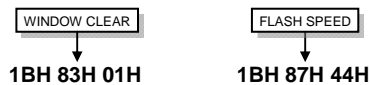
SCROLL

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: -

| SELECT WINDOW 1 | SELECT FONT | DEFINE WINDOW CO-ORDINATES |

1BH 80H

1BH 82H 10H 10H 67H 2FH

1BH 98H 0BH

1BH 90H   02H   01H   "Scrolling" 0DH "Text Message" 00H

| SCROLL | LEFT | ONCE | SCROLL TEXT | NEW LINE | SCROLL TEXT | TEXT END |

Scrolling horizontal text data.
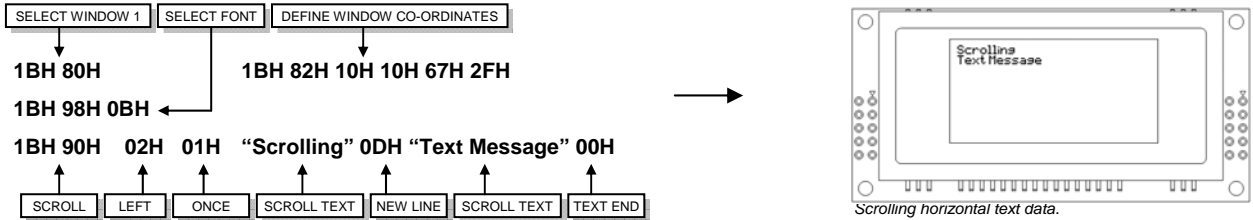
PATTERNS

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.
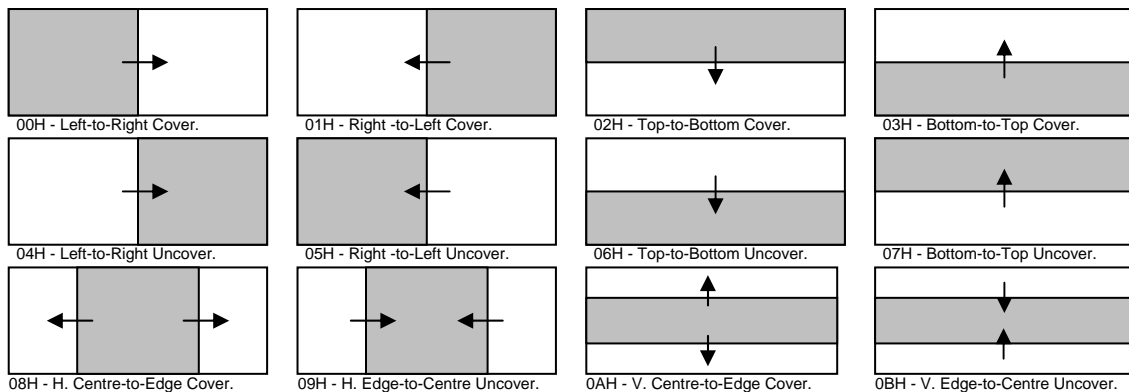
Custom Pattern Data.

Pattern example: -

| SELECT WINDOW 1 | SHOW WINDOW |
| ENABLE PATTERN | DEFINE WINDOW CO-ORDINATES |

1BH 80H

1BH 82H 00H 00H 7FH 3FH

1BH 83H 03H   **1BH 84H**

1BH 8EH | DEFINE PATTERN | 32-BYTE PATTERN DATA |

86H FCH FEH 86H 7EH 86H 00H 86H FCH 86H 86H FEH 86H 7EH 86H 00H

86H FCH FEH 86H 7EH 86H 00H 86H FCH 86H 86H FEH 86H 7EH 86H 00H

| INVERT |

1BH 8FH 08H

Define custom pattern.

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: -

00H - Left-to-Right Cover.   01H - Right -to-Left Cover.   02H - Top-to-Bottom Cover.   03H - Bottom-to-Top Cover.

04H - Left-to-Right Uncover.   05H - Right -to-Left Uncover.   06H - Top-to-Bottom Uncover.   07H - Bottom-to-Top Uncover.

08H - H. Centre-to-Edge Cover.   09H - H. Edge-to-Centre Uncover.   0AH - V. Centre-to-Edge Cover.   0BH - V. Edge-to-Centre Uncover.

Wipe examples: -

SELECT WINDOW 1

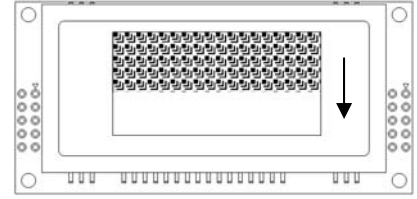**1BH 80H**

1BH 83H 03H

ENABLE WIPE PATTERN

DEFINE WINDOW CO-ORDINATES

1BH 82H 00H 00H 7FH 3FH

1BH 88H 02H
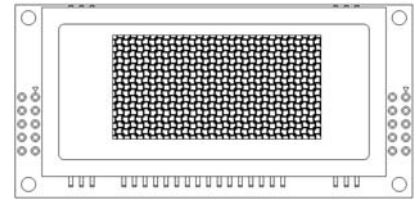
START WIPE DOWN EFFECT

Wipe down effect with pattern enabled.

CHANGE WIPE PATTERN

1BH 8DH 07H

LOCK PATTERN TO BOTTOM OF WINDOW

1BH 8FH 06H

Change pattern.

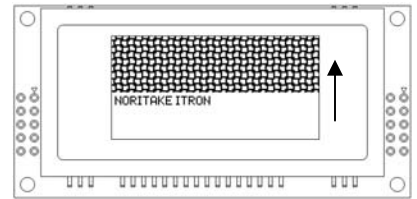SELECT 7x15 FONT

1BH 98H 0BH

SET CURSOR

10H 22H 20H
10H 17H 30H

1BH 88H 07H

UNCOVER WIPE EFFECT

DISPLAY TEXT

"Welcome to"
"Noritake Itron"

Uncover hidden text.

ADDITIONAL FONTS

Cyrillic 5x7 & 10x14 Font

Standard 7x15 Font

20x28 Font