



GHLCD 3224C

Serial/I²C Graphic LCD

Introduction

With GHLCD, outputting graphics and text to the LCD is a walk in the park. In most cases, to use a graphic display, the user must study the graphic LCD controller and must spend a lot of time for expressing the graphics and text.

The GHLCD was created to solve all these types of problems. By embedding an LCD control module, the user can simply send commands through RS232C (or I²C) to print graphics and text. There is also a memory space for BMP files.

To print graphics and text, an RS232C communication is required. Through RS232C, the LCD can easily be controlled by the user using simple commands.

For sending data to the LCD, the user may use a CUBLOC, PC, or other microcontrollers.

The GHLCD has many features including a Backlight & Contrast Control, dots, lines, boxes, and circles to create GUI friendly applications. With an internal RAM for fonts, the user may create custom characters.

Main Features

- Dot, Line, Circle, Ellipse, Box, Pop, Paste, and Paint commands supported
- 2400 ~ 115200 Baud rates are supported.
- I²C interface are supported.
- ±12V (RS232C) and 5V (TTL) are supported.
- 4 Built-in Fonts
- Separate Text & Graphic Layers (Total of 3 Layers)
- The Text Layer supports scroll, cursor, inversion, underline, and bold.
- The Text Layer uses Character-based pixels and the Graphic Layer uses Dot-based pixels
- Push/Pop/Paste commands supported. (Like Copy & Paste in Windows)
- Display BMP Files
- Set the dot size for Lines, Circles, Boxes, Ellipses.
- Software controlled Backlight and Contrast
- Control Box Program to test the LCD.

GHLCD Hardware

There are 2 RS232C outputs, 5V and 12V. The PC uses ±12V RS232C signals while the 5V signal can be used for CUBLOC or other micro-controllers. The 5V is good for very short distances(less than 3 ft.). For longer distances or noisy environments, please use the 12V level connections.



The 12V level RS232C has 4 pin connector whereas the 5V level RS232C has 3 pin connector. With the 5V RS232C, TX is not used. GHLCD usually only receives data, so TX is not really required.

The graphic LCD also supports I2C interface. Using CUBLOC's CuNET (or I2C) to GHLCD's I2C port, you can use native commands to easily control the LCD.

CUBLOC native commands include such basic commands as CLS, LOCATE, PRINT, CIRCLE, LINE, and BOX.

The following is a sample program to produce the text on the LCD:

<pre> Const Device = CB280 Set Display 1,0,1,30 Dim a As Integer a = 0 Cls Contrast 400 Glocate 1,80 Font 8,1 GPrint "GHLCD" Glocate 180,120 GPRINT "Demo" Do Locate 0,0 Print Dec a,Cr Incr a Delay 10 Loop </pre>	
---	--

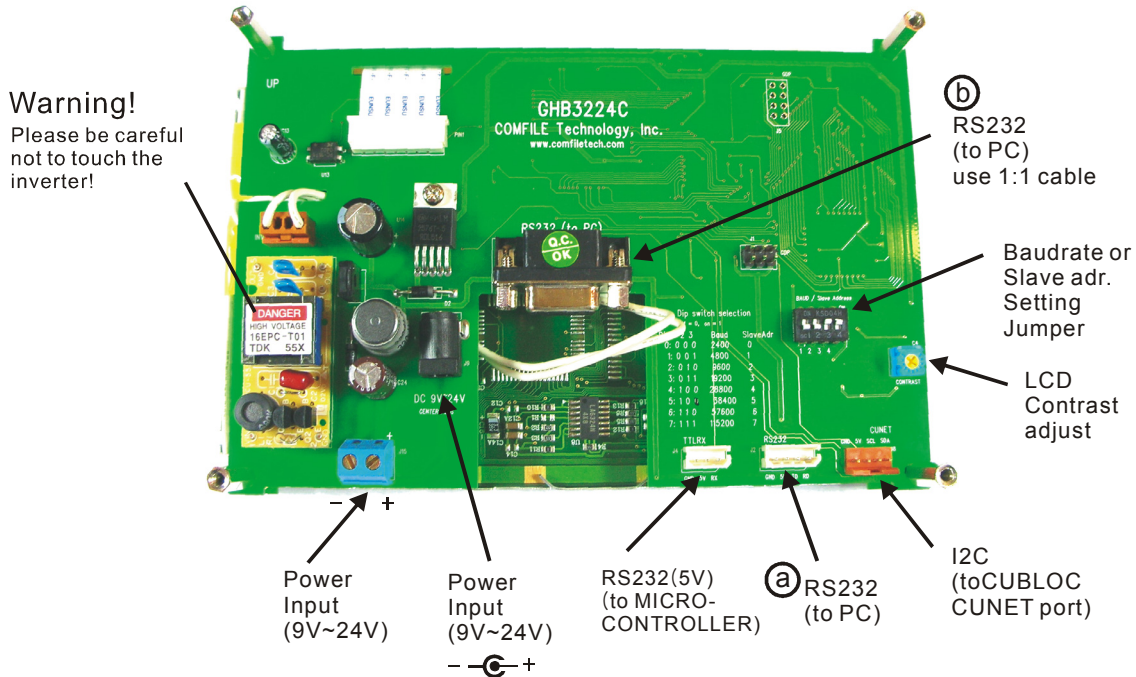
For power, please use at least 1A of current for 9-12V adaptor or SMPS. For 12-24V, please use at least 500mA of current. The following is power consumed by the GHLCD. (The actual power consumed may vary by the temperature).

Input	Backlight ON	Backlight OFF
9V	445mA	113mA
12V	323mA	84mA
24V	149mA	42mA



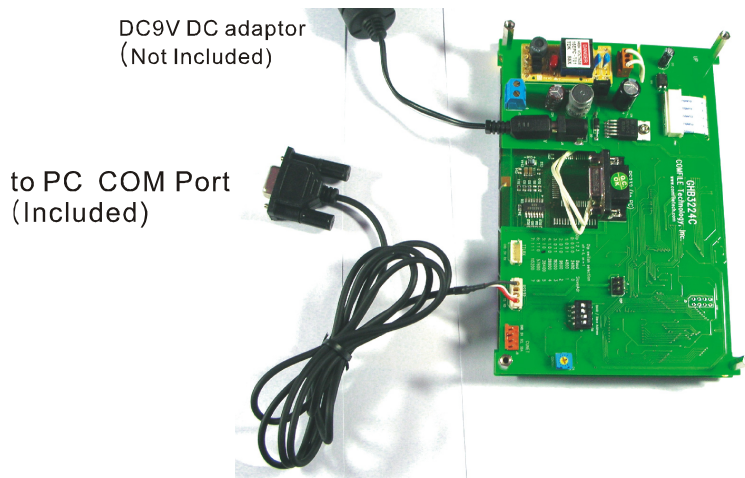
Connection of GHLCD

On the backside of GHLCD, there are various optional connectors and switches so the user is able to interface with his/her controller.



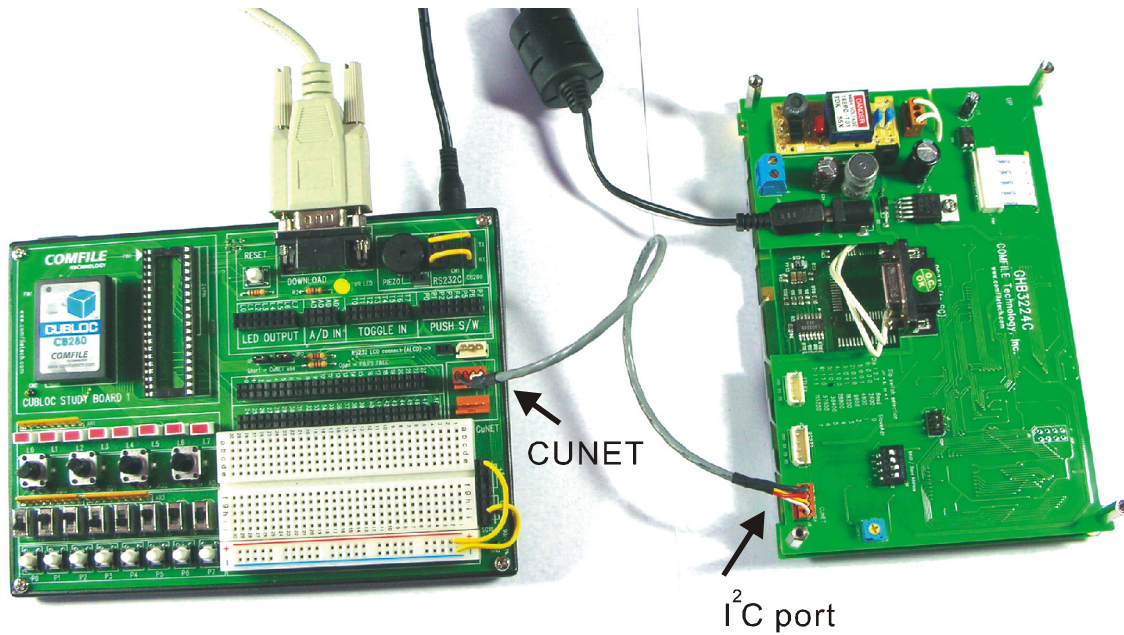
(a) and (b) are RS232C ports. Either RS232C port may be used.

(a) is an RS232C PORT for connecting to PC using the 4-pin to serial cable provided w/ GHLCD. (b) is an RS232C serial port for connecting to PC using a standard serial cable. You are free to choose which port to use to communicate to the GHLCD.

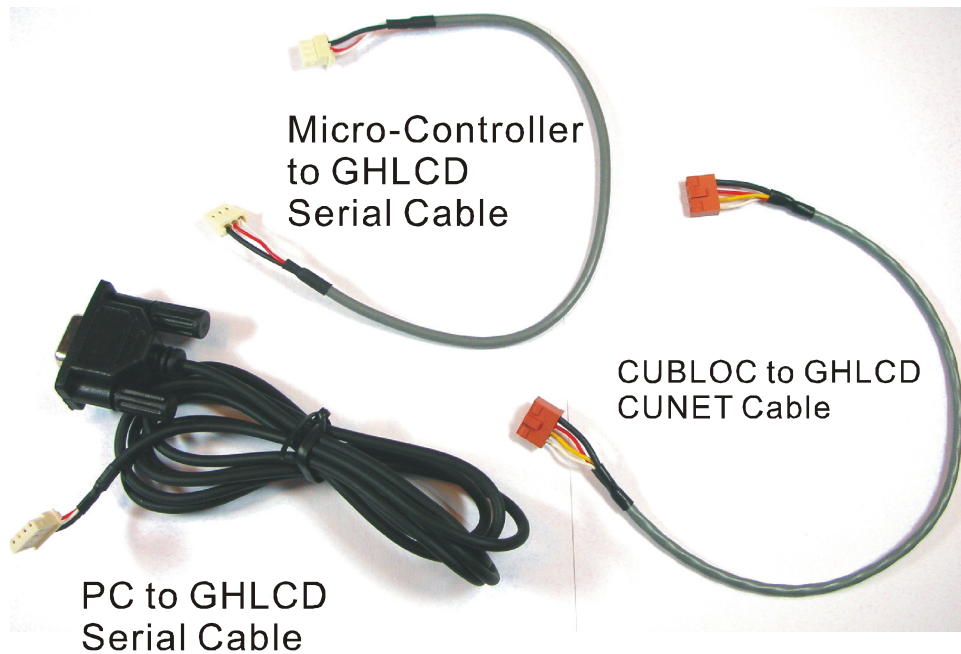




When connecting to CUBLOC, you can simply use the CUNET (I2C) cable provided with the GHLCD.



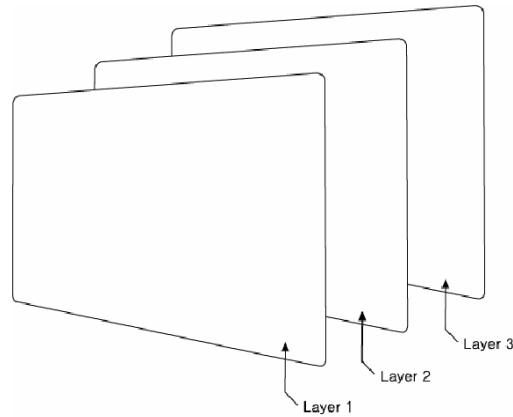
There are 3 cables that are included with the GHLCD. Please make sure you have all 3.





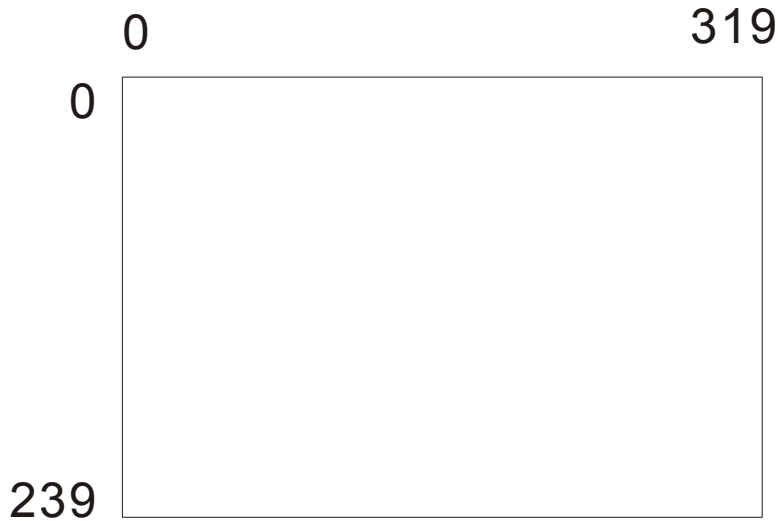
Layers of GHLCD

There are 3 layers. Each layer can display graphics. Each layer can be independently controlled. It's as though there are 3 LCD screens in one. By using these 3 layers, the user has more flexibility and will be able to create eye-popping applications. The user can use one of the layers as background layer while using the other two for animation, text, and etc... Text can only be written to Layer 1. Graphics can be displayed on any on the 3 layers. The fact default graphic layer at power ON is Layer 2.



Graphic Layer X, Y Axis

There are 0 to 319 for X Axis and 0 to 239 for Y Axis.



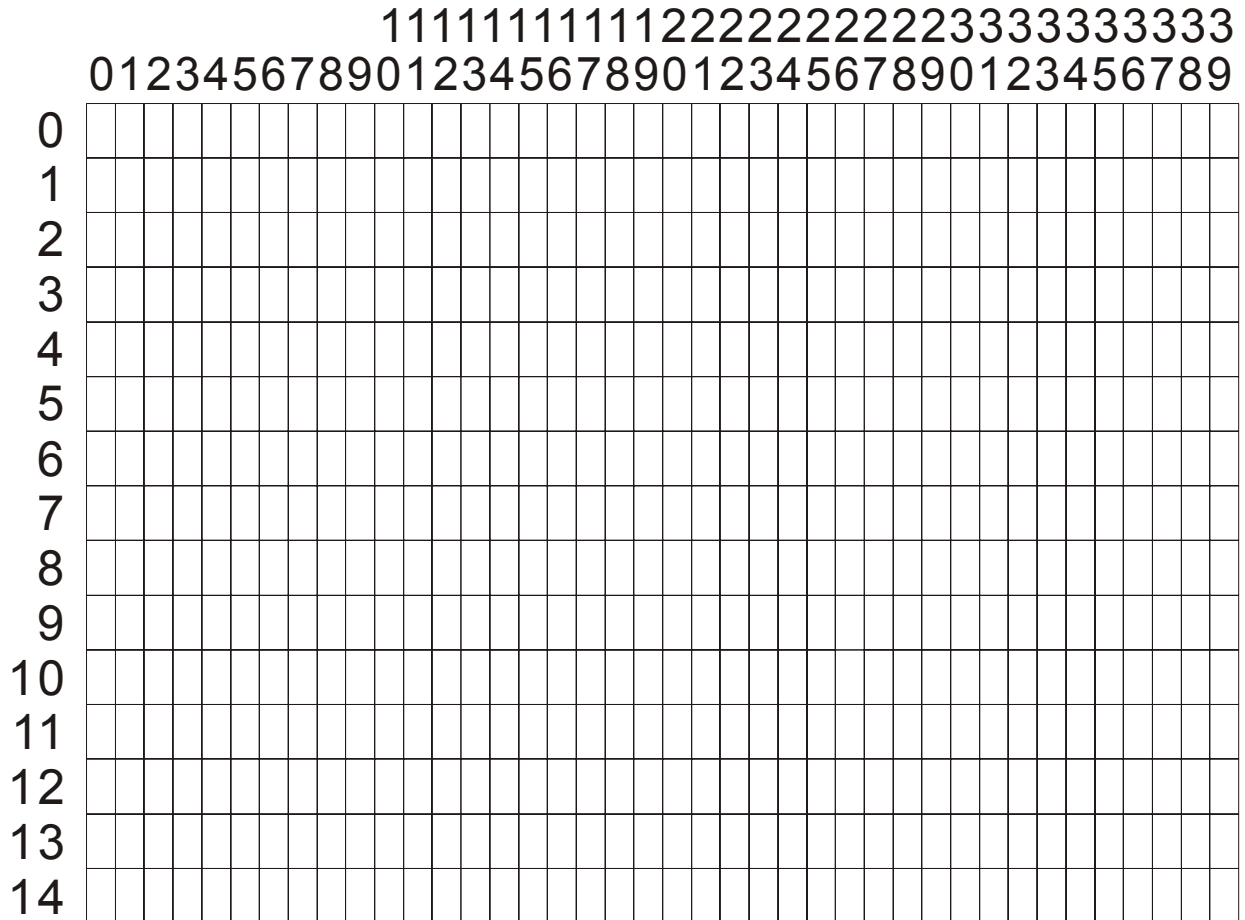


X and Y Axis

In GHLCD, text layer and graphic layer use different axis. Text layer uses character units whereas graphic layer uses pixel units.

Text Layer X, Y Axis

There are 0 to 39 for X Axis and 0 to 14 for Y Axis. Each character is 8 x 16 pixels in size.



● Use this page as template

In case of text layer, when a character is printed to (39,14), the screen will automatically scroll one line.



How to communicate with GHLCD

The user can choose one of the following to communicate with the GHLCD:

1. Connect CUBLOC to GHLCD using the CUNET (I2C) and use easy-to-use native commands.
2. Connect PC to GHLCD using either RS232C port or the RS232C 4-pin port.
3. Connect a CUBLOC or a Micro-controller to the GHLCD using the 3-pin TTL connection wires

Using one of the three ways, the user can easily communicate with the GHLCD. When using CUBLOC and CUNET, the user will be able to take advantage of the native library support by the RTOS on CUBLOC. When using other micro-controller or PC, the user can refer to raw hex commands shown on the bottom of each command in this manual.

To print to GHLCD, you need to send an ESC character (0x1B), then two characters of a graphic command, and then reset of the parameters. If you simply send characters(in ASCII) without the ESC character and graphic command, the GHLCD will simply print the data to the current cursor position on the Text Layer.

<System Setting Commands>

Set Layer

ESC	P M	layer1	layer2	layer3
1byte	2byte	1byte	1byte	1byte

Parameters : layer1, layer2, layer3 = (0=off, 1=on, 2=2Hz Flash, 3=16Hz Flash)

Operation : Each Layer can be turned off, on, or flashed at 2Hz or 16Hz.

Default values are: layer1 = 1, layer2 = 1, layer3 = 0

Send Example : (Hex) 1B 50 4D 01 01 00

Tips : Set Layer 3 off and use as buffer. You can draw to the layer 3 and then turn on layer 3, which will cause no delay in printing the graphics.

Set Graphic Layer

ESC	P G	n
1byte	2byte	1byte

Parameters : n = 1,2, or 3

Operation : Set Graphic Layer. At power ON, Graphic layer is set to Layer 2. LINE, CIRCLE, BOX, PAINT, and other graphic commands are displayed onto the set graphic layer.

Send Example : (Hex) 1B 50 47 02



Overlap Mode Settings

ESC	OM	n
1byte	2byte	1byte

Parameters: n= (0=OR, 1=AND, 2=XOR)

Operation : Set the logical operation between layer 1 and layer 2. When you are displaying both text and graphics to Layers 1 and 2, respectively, you will sometimes have overlapping between text and graphics. When this happens, you can either to have the overlapped area to be printed as is (0=OR), to print as clear (1=AND), or you can have it printed inverted (2=XOR).

Send Example : (Hex) 1B 4F 4D 00

Changing Contrast

ESC	VE	n(low)	n(high)
1 byte	2 bytes	1 byte	1 byte

Parameters : n = 0 to 1024. N is a 16 bit value. Please send the LOW byte FIRST, then the HIGH byte.

Operation : Control the contrast of the LCD. The adjusted value is stored in EEPROM of the LCD, therefore there is no need to set contrast at power-ON.

Send Example : (Hex) 1B 56 45 45 01

Tips : You can also use a screw driver to adjust the Contrast variable resistor located near the lower-right side on the back of the GHLCD. Using the software contrast control command and using the hardware resistor, the user can optimize his/her GHLCD to the maximum.

BACK LIGHT ON/OFF

ESC	BL	N
1byte	2byte	1byte

Parameters : n = (0=OFF, 1=ON)

Operation : Set LCD Backlight ON/OFF

Send Example : (Hex) 1B 42 4C 00 01

WRITE MODE

ESC	WM	n
1byte	2byte	1byte

Parameters : n = (fast, 1=slow)

Operation : You can set the LCD to print in fast mode or slow mode. Fast mode can be used in most cases. Use slow mode when you have trouble with vibrating screen.

Send Example : (Hex) 1B 57 4D 01



Setting FONT

ESC	F O	f1	f2
1 byte	2 bytes	1 byte	1 byte

Parameters : f1 = 0 to 8 (Font, Size)

f2 = (0 = Fixed Width Font, 1 = Variable Width Font)

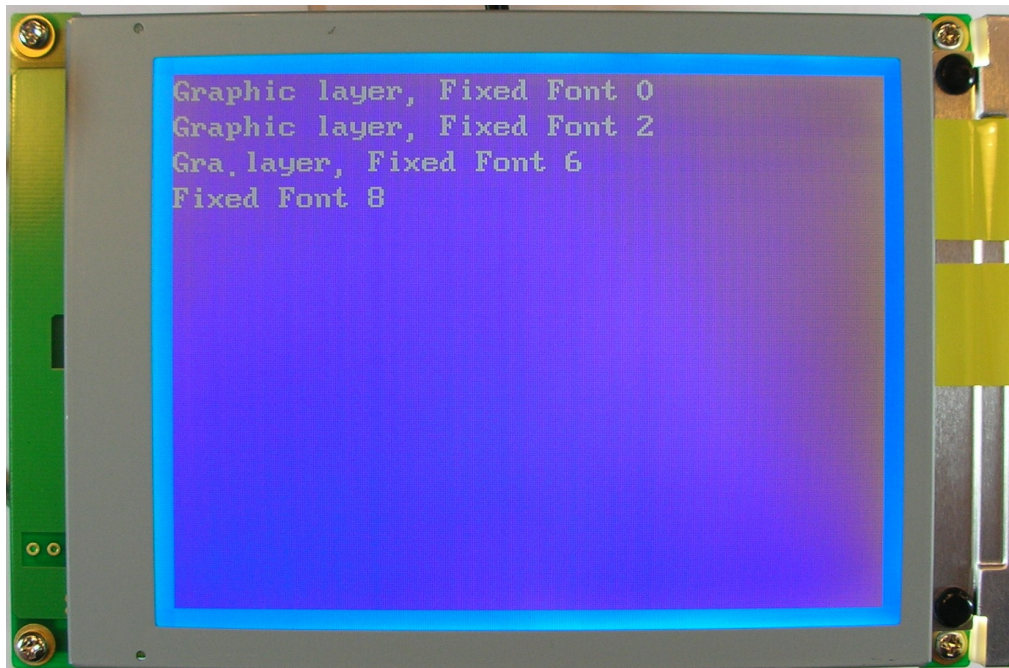
Sizes

0,1	5x16 Arial
2,3,4,5	8x16 Times New Roman
6, 7	12x24 Arial
8	24x48 Arial

Send Example : (Hex) 1B 46 4F 04 01

The GHLCD has separate fonts for the Text Layer and the Graphic Layer. In the Text Layer, only 16x16 pixel characters may be used for text. In the graphic layer, all 4 different fonts and sizes may be used.

The following is an example of Text Layer font. Although limited to one font, the Text Layer text is much faster than the Graphic Layer text.

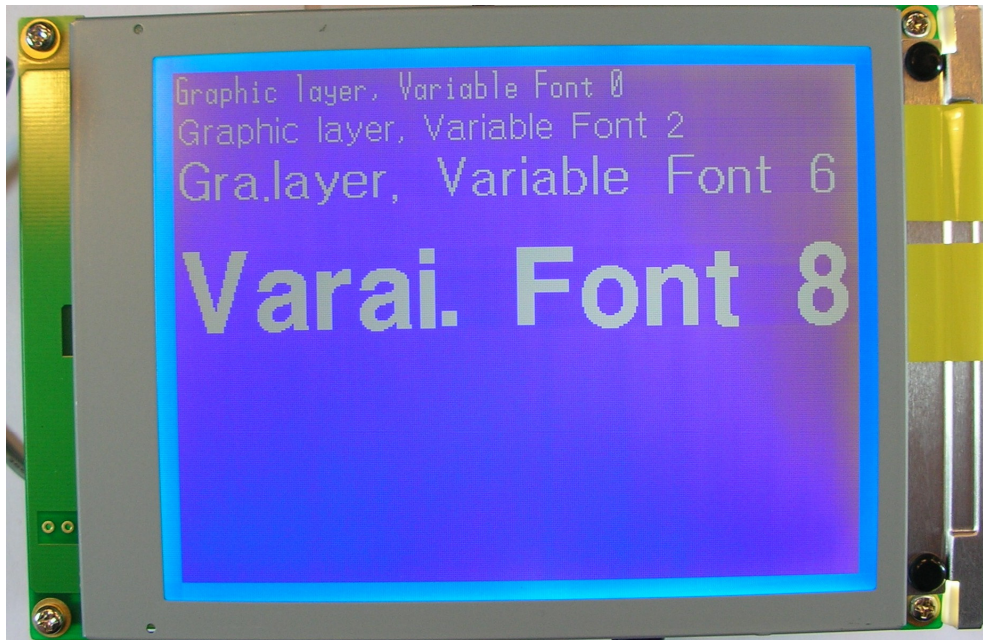




The following is an example of **fixed** width fonts in the Graphic Layer.



The following is an example of **variable** width fonts in the Graphic Layer.





Set Font Style

ESC	F S	Bold	Inverse	Underline
1byte	2byte	1byte	1byte	1byte

Parameters : Bold = (0=None, 1=None, 2=Bold, 3=Bold)

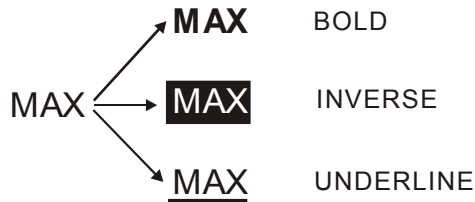
Inverse = (0=None, 1=Inverse)

Underline = (0=None, 1=Underline)

Operation : Bold, Inverse, or Underline your Font.

Default is set to None.

Send Example : (Hex) 1B 46 53 00 00 00



CLEAR

ESC	C L	n
1byte	2byte	1byte

Parameters : n = (0=Clear All layers, 1=Clear Layer 1, 2=Clear Layer2, 3=Clear Layer3)

Operation : You can clear all layers, or specify the layer you want to clear.

Send Example : (Hex) 1B 43 4C 00



<Text Layer Commands>

LOCATE

ESC	L O	x	y
1byte	2byte	1byte	1byte

Parameters : x = (x axis value, y=y axis value)

Operation : You can set the cursor of the text layer with this command. When CLEAR command is sent, the cursor is automatically set to 0,0.

Send Example : (Hex) 1B 4C 4F 05 05

Cursor ON/OFF

ESC	C S	n
1byte	2byte	1byte

Parameters : n (0=off, 1=on)

Operation : Turn Text Layer Cursor ON/OFF.

Default is OFF.

Send Example : (Hex) 1B 43 53 01

Set Cursor Mode

ESC	C M	n
1byte	2byte	1byte

Parameters : n = (0=Box, 1=Underline)

Operation : Set the Cursor type to be a box or an underline.

Default is Box.

Send Example : (Hex) 1B 43 4D 00



box



underline



<Graphic Layer Commands>

- The parameters in graphic layer commands are in units of Words. You must send the LOW byte first, then the HIGH byte.

LINE

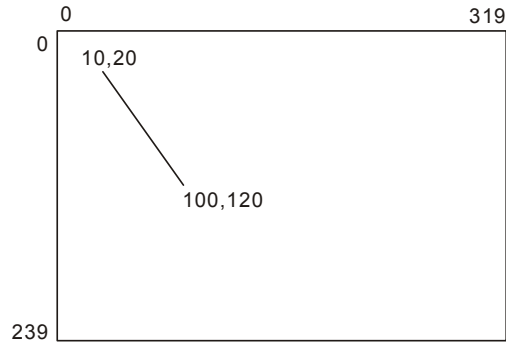
ESC	LI	X1	Y1	X2	Y2
1byte	2byte	2byte	2byte	2byte	2byte

Parameters : X1, Y1=Starting X,Y coordinate, X2, Y2=Ending X,Y coordinate

Operation : Draw a line starting from X1, Y1 to X2, Y2.

Send Example : When display Line(10,20) (100,120)

(Hex) 1B 4C 49 0A 00 14 00 64 00 78 00



LINETO

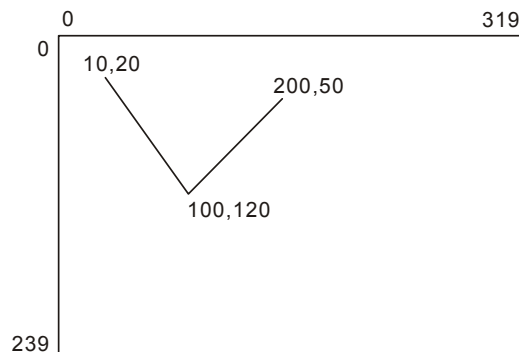
ESC	LT	X	Y
1byte	2byte	2byte	2byte

Parameters : X, Y=Ending X,Y coordinate

Operation : Continue to draw line from the last ending X,Y coordinate. You can use this Lineto command to continue drawing lines without having to use the starting X,Y coordinate every time.

Send Example : When drawing a line to (200,50)

(Hex) 1B 4C 54 C8 00 32 00





BOX

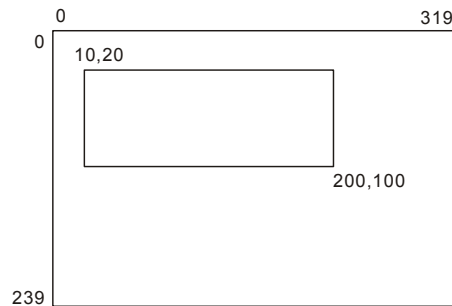
ESC	B X	X1	Y1	X2	Y2
1byte	2byte	2byte	2byte	2byte	2byte

Parameters : X1, Y1=Starting X,Y coordinate, X2, Y2=Ending X,Y coordinate

Operation : Draw a box on the Graphic Layer using X1,Y1 as the most left-upper point and X2, Y2 as the most right-lower point.

Send Example : When Displaying a box from (10,20) to (200,100)

(Hex) 1B 42 58 0A 00 14 00 C8 00 64 00



BOXFILL

ESC	B F	X1	Y1	X2	Y2	Logic
1byte	2byte	2byte	2byte	2byte	2byte	1byte

Parameters : X1, Y1=Starting X,Y coordinate, X2, Y2=Ending X,Y coordinate, logic =(0=OR, 1=AND, 2=XOR)

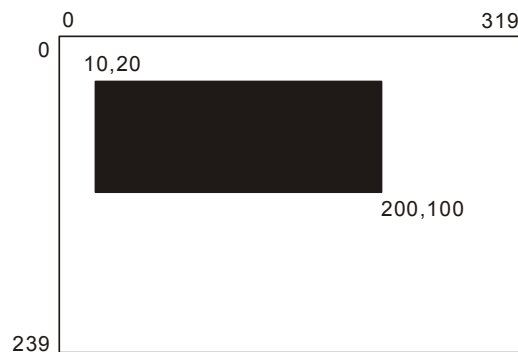
Operation : Fill a box on the Graphic Layer using X1,Y1 as the most left-upper point and X2, Y2 as the most right-lower point.

Logic will determine the overlapped area.

- 0 will do a logical OR operation for overlapped area
- 1 will do a logical AND operation for overlapped area
- 2 will do a logical XOR operation for overlapped area

Send Example : When filling a box from (10,20) to (200,100)

(Hex) 1B 42 46 0A 00 14 00 C8 00 64 00 00





Tips : When using BOXFILL command, you can use the logical XOR to invert itself. You can make flashing effects easily with this command.

BOX CLEAR

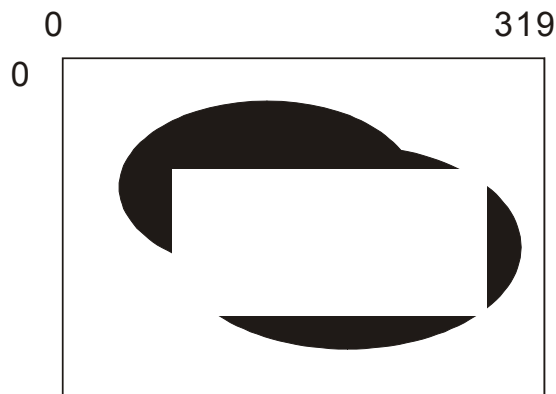
ESC	B C	X1	Y1	X2	Y2
1byte	2byte	2byte	2byte	2byte	2byte

Parameters : X1, Y1=Starting X,Y coordinate, X2, Y2=Ending X,Y coordinate

Operation : Clear a box on the Graphic Layer using X1,Y1 as the most left-upper point and X2, Y2 as the most right-lower point.

Send Example : When Displaying a box from (10,20) to (200,100)

(Hex) 1B 42 43 0A 00 14 00 C8 00 64 00





CIRCLE

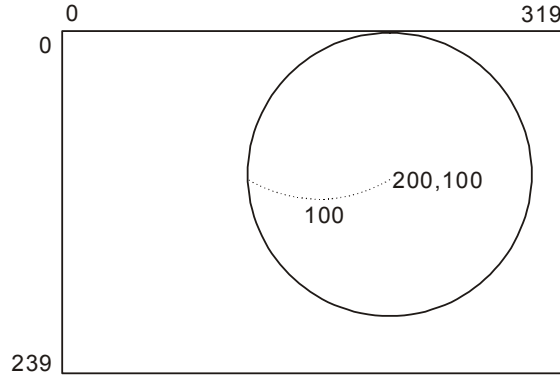
ESC	C I	X	Y	R
1byte	2byte	2byte	2byte	2byte

Parameters : X, Y=Center, R=Radius

Operation : Draw a circle with X,Y center coordinate and radius R.

Send Example : Draw a circle with (200,100) center and radius of 100

(Hex) 1B 43 49 C8 00 64 00 64 00



CIRCLE FILL

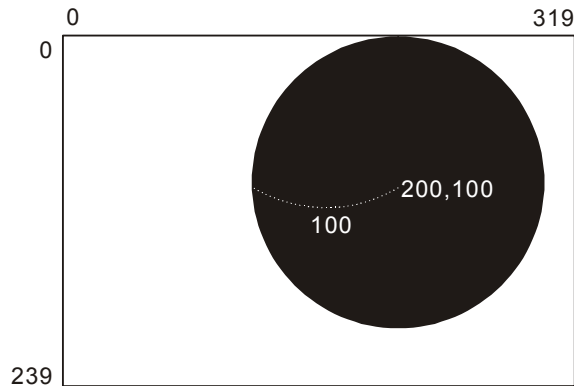
ESC	C F	X	Y	R
1byte	2byte	2byte	2byte	2byte

Parameters : X, Y=Center, R=Radius

Operation : Fill a circle with X,Y center coordinate and radius R.

Send Example : Fill a circle with (200,100) center and radius of 100

(Hex) 1B 43 46 C8 00 64 00 64 00





ELLIPSE

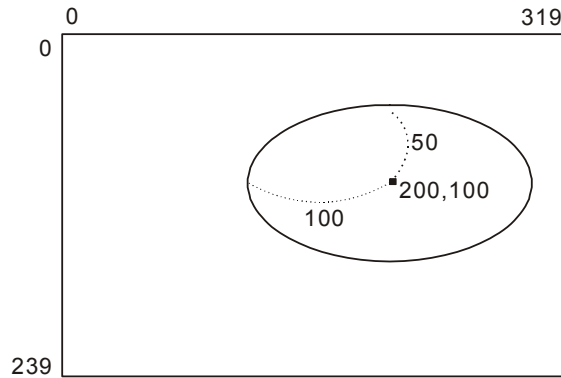
ESC	E L	X	Y	R1	R2
1byte	2byte	2byte	2byte	2byte	2byte

Parameters : X, Y=Center, R1=Horizontal Radius, R2=Vertical Radius

Operation : Draw an ellipse with X,Y as center and R1 and R2 as horizontal and vertical radius, respectively.

Send Example : Draw an ellipse with X, Y center and R1 and R2 as radiuses.

(Hex) 1B 45 4C C8 00 64 00 64 00 32 00



ELLIPSE FILL

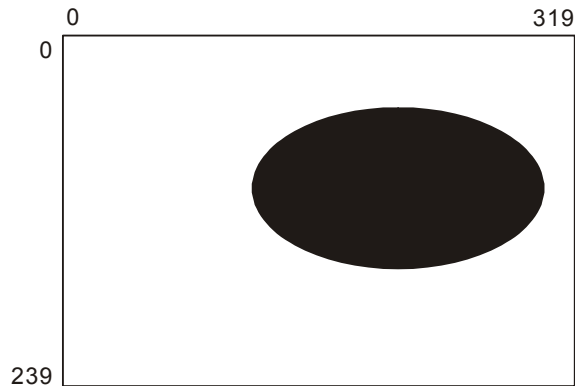
ESC	E F	X	Y	R1	R2
1byte	2byte	2byte	2byte	2byte	2byte

Parameters : X, Y=Center, R1=Horizontal Radius, R2=Vertical Radius

Operation : Fill an ellipse with X,Y as center and R1 and R2 as horizontal and vertical radius, respectively.

Send Example : Fill an ellipse with X, Y center and R1 and R2 as radiuses.

Send Example : (Hex) 1B 45 4C C8 00 64 00 64 00 32 00





Displaying text to the Graphic Layer is much flexible than displaying text to the Text Layer. The user is allowed to display text to any pixel location on the screen using the Graphic Layer. On the other hand, the user is limited to 16x16 pixel locations when displaying text to the Text Layer. The Graphic Layer text print command is **Gprint**. The user also has the option of using the text print command **Dprint**, which allows the user to automatically erase existing graphics or text beforehand. **Dprint** is also much faster than **Gprint**, comparable to regular **print** command. **Glocate** may be used in place of **Locate** to specify pixel location before using print commands.

GRAPHIC LOCATE

ESC	GL	X	Y
1byte	2byte	2byte	2byte

Parameters : X, Y = New X, Y coordinates for Graphic Layer

Move to a specific x, y coordinate pixel location on the screen for Graphic Layer.

Send Example : (Hex) 1B 47 4C C8 00 64 00

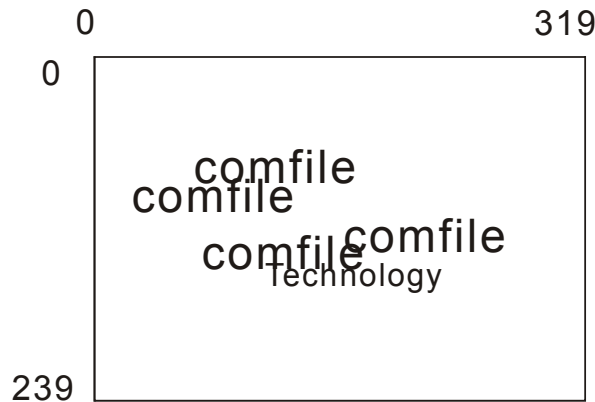
GRAPHIC STRING PRINT

ESC	PR	String	end code '0'
1byte	2byte	N byte	1byte

Parameters : string = String to print to Graphic Layer, end code = (End of Line) '0'

Graphic layer 문자표시 명령으로, 기존의 바탕화면에 겹쳐서 text를 표시합니다. String끝에는 반드시 Null character를 붙여주어야합니다. 이 명령은 Graphic layer상에서 모든 font를 표시할 수 있고, OF명령으로 지정한 Offset이 적용되어 표시됩니다.

Send Example : (Hex) 1B 50 52 BE C6 0D 0A 00





GRAPHIC DIRECT STRING PRINT

ESC	D R	string	end code '0'
1byte	2byte	N byte	1byte

Parameters : string = String to print to Graphic Layer, end code = (End of Line) '0'

Print text overlapping current graphics or text. When displaying values that change in real-time such as temperature, this command can be used for faster display speed.

The only difference from Gprint is that the X coordinates must be multiple of 8. Only font sizes 2 and 6 with Fixed Width font may be used.

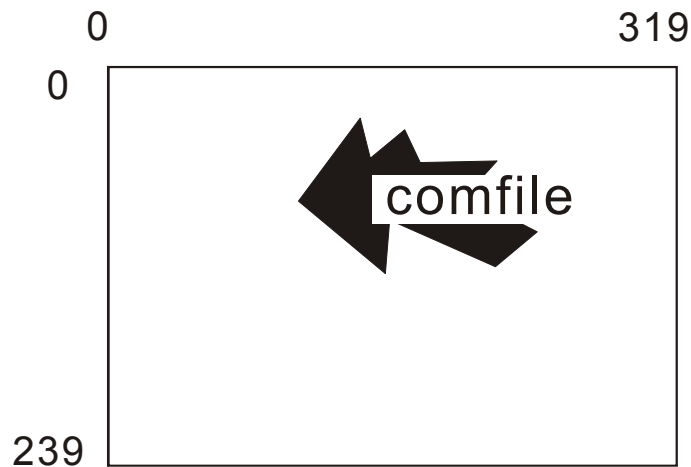
Font 2,0

Font 6,0

Only two fonts are available for this command.

Offset set by the Offset command does not affect the Dprint command.

Send Example : (Hex) 1B 44 52 BE C6 0D 0A 00





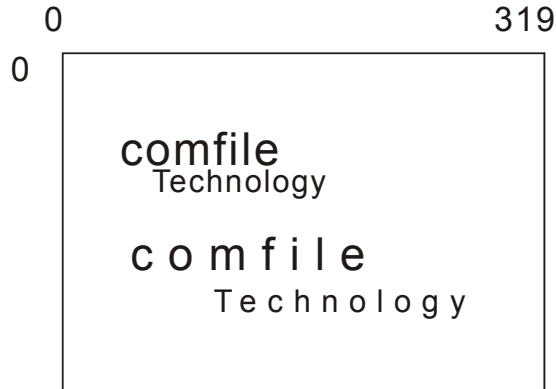
OFFSET

ESC	OF	x_offset	y_offset
1byte	2byte	1byte	1byte

Parameters : x_offset = x coordinate offset, y_offset = y coordinate offset

Operation : Use offset to put offset for printing text to the Graphic Layer. Default value is 0.

You can use x_offset and y_offset to fit more characters in your screen or to balance the test.



PSET

ESC	P S	X	Y
1byte	2byte	2byte	2byte

Parameters : X, Y = X, Y Coordinates

Operation : Draw one dot.

Send Example : (Hex) 1B 50 53 C8 00 64 00

SET COLOR

ESC	S C	color
1byte	2byte	1byte

Parameters : color = 0 or 1 (0 for Clear and 1 for Blue)

Send Example : (Hex) 1B 53 43 00



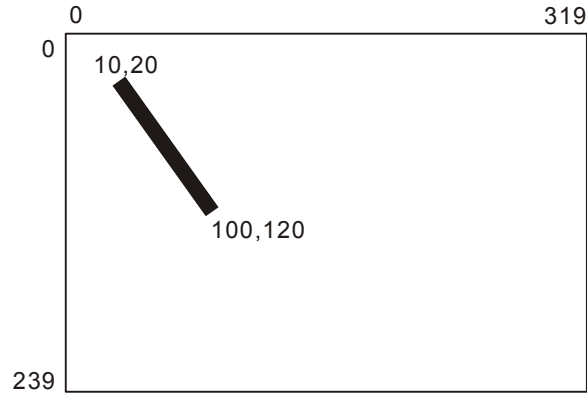
DOT SIZE

ESC	D S	n	s
1byte	2byte	1byte	1byte

Parameters : n = Dot size between 0 and 255, s = Dot type (0=square, 1=circle)

Operation : Set the size of the dot.

Send Example : (Hex) 1B 44 53 01 01



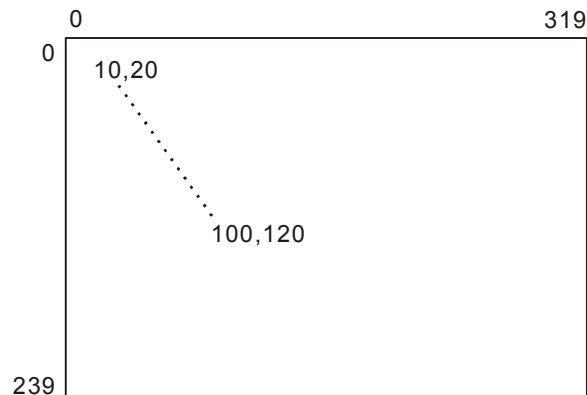
LINE STYLE

ESC	L S	n
1byte	2byte	1byte

Parameters : n = Dot space

Send Example : (Hex) 1B 4C 53 03

Example : Set to 5 and Draw Line from (10,20) to (100,120)





PAINT

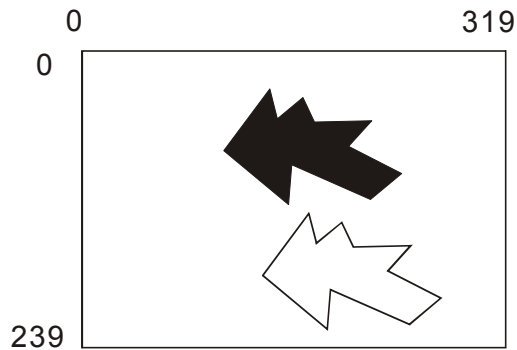
ESC	PA	X	Y
1byte	2byte	2byte	2byte

Parameters : X, Y = X, Y Coordinates

Operation : Paint within the enclosed space.

Send Example : (Hex) 1B 50 41 C8 00 64 00

If the enclosed space has even a little bit of hole, the whole screen will be painted. TO use this command, Color must be set to 1. (Color 1)



ARC

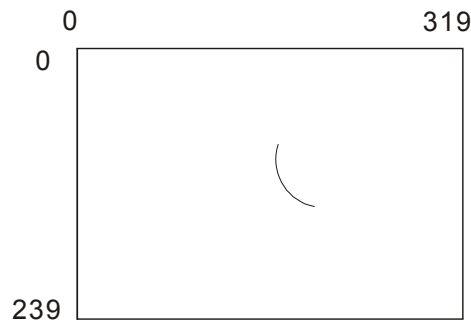
ESC	AR	X	Y	R	START	END
1byte	2byte	2byte	2byte	2byte	2byte	2byte

Parameters : X, Y=Center, R=Radius, START=Starting angle, END=Ending Angle

The ARC command can be used to draw arcs.

Example : X=200, Y=50, R=100, START=10, END=200

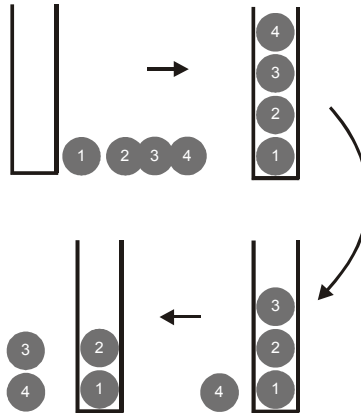
Send Example : (Hex) 1B 41 52 C8 00 32 00 64 00 0A 00 C8 00





<Screen push, pop, paste Commands>

The GHLCD has 60KB Stack for temporary storage of graphic data. By using stack, the user can store part of the screen in the RAM of GHLCD comparable to copy and paste function of normal text editors. Stack operates in LIFO (Last In First Out). The last item pushed will be popped first.

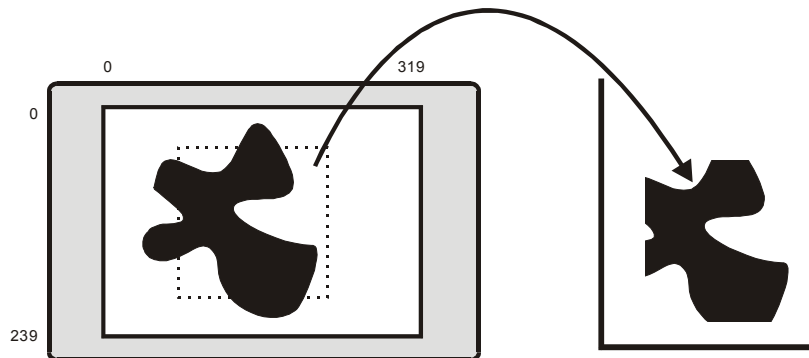


SCREEN PUSH

ESC	P U	X1	Y1	X2	Y2	Layer
1byte	2byte	2byte	2byte	2byte	2byte	1byte

Parameters : X1, Y1 = Starting X, Y coordinates, X2, Y2 = Ending X,Y coordinates, layer = Layer #

Send Example : (Hex) 1B 50 55 C8 00 32 00 64 00 0A 00 02





SCREEN POP

ESC	P O	X	Y	layer	logic
1byte	2byte	2byte	2byte	1byte	1byte

Parameters : X, Y = X, Y coordinates to POP, layer = Layer #

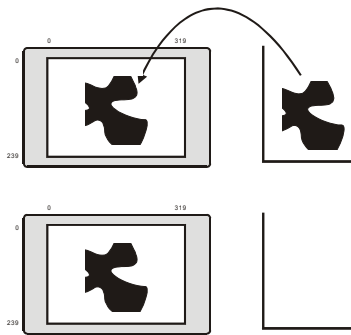
logic=0 : OR

logic=1 : AND

logic=2 : XOR

logic=3 : Clear screen before popping

Send Example : (Hex) 1B 50 4F C8 00 32 00 02 00



SCREEN PASTE

ESC	P P	X	Y	layer	logic
1byte	2byte	2byte	2byte	1byte	1byte

Parameters : X, Y = X, Y coordinates to POP, layer = Layer #

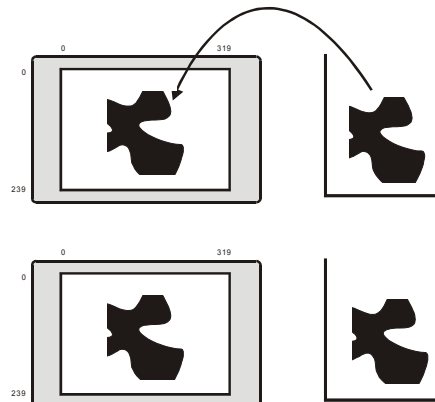
logic=0 : OR

logic=1 : AND

logic=2 : XOR

logic=3 : Clear screen before popping

Send Example : (Hex) 1B 50 50 C8 00 32 00 02 00





HIGH SPEED PUSH

ESC	H U	X1	Y1	X2	Y2	layer
1 byte	2 byte	2 byte	2 byte	2 byte	2 byte	1 byte

Parameters : X1, Y1 = Starting X, Y coordinates, X2, Y2 = Ending X, Y coordinates, layer = Layer #

The high speed commands are fast as commands are processed in byte units. But it lacks the capability to do logical operations. Please use this command for saving and restoring parts of the screen. The High Speed X coordinates must be set from 0 to 39. This is byte units of the 0 to 319 pixels.

Send Example : (Hex) 1B 48 55 01 00 32 00 03 00 0A 00 02

HIGH SPEED POP

ESC	H O	X	Y	Layer
1 byte	2 bytes	2 bytes	2 bytes	1 byte

Parameter : X, Y = location to POP to, Layer = Layer number

X must be between 0~39 (byte units)

Operation : Pop from stack. This command will erase the background before popping.

Send Example : (Hex) 1B 48 4F 01 00 32 00 02

HIGH SPEED PASTE

ESC	H P	X	Y	Layer
1 byte	2 bytes	2 bytes	2 bytes	1 byte

Parameter : X, Y = Location to POP to, Layer = Layer number,

X must be between 0~39 (byte units)

Send Example : (Hex) 1B 48 50 01 00 32 00 02



<BMP Commands>

With GHLCD, BMP files can be directly printed to the LCD or saved in the flash memory before displaying the BMP file.

BMP PRINT

ESC	B P	X	Y	n	layer
1byte	2byte	2byte	2byte	1byte	1byte

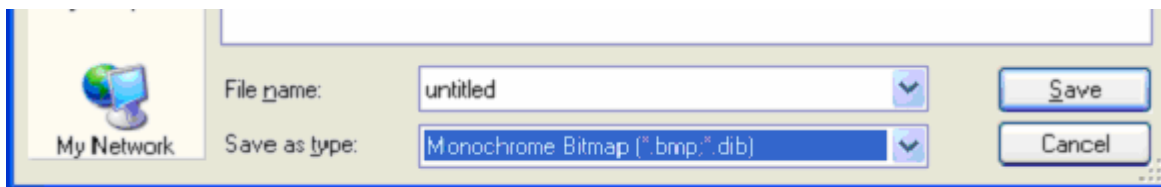
Parameters : X, Y = X, Y coordinates to print BMP file.

n = BMP file number in the flash memory

layer = Layer to print the BMP file(1~3)

Send Example : (Hex) 1B 42 50 C8 00 32 00 01 02

This command will print the BMP file stored in the Flash memory of the GHLCD. To use this command, the user must first download BMP files to the GHLCD using a PC. The program **BMP Download** can be used to download up to about 100 full pages of BMP files to the GHLCD's Flash memory. For the GHLCD model GHB3224C, only black and white images are supported. Please make sure to save your graphic file as a **Monochrome Bitmap AND less than 320x240 pixels**.



BMP Direct PRINT

ESC	B D	X	Y	Data
1byte	2byte	2byte	2byte	BMP file

Parameters : X, Y = X, Y coordinates to print BMP file.

data = BMP file

This command will print the BMP file directly to the GHLCD. This command can be used when Flash memory capacity is full.



DEFINE CHARACTER

ESC	D F	code	Data
1byte	2byte	2byte	2byte

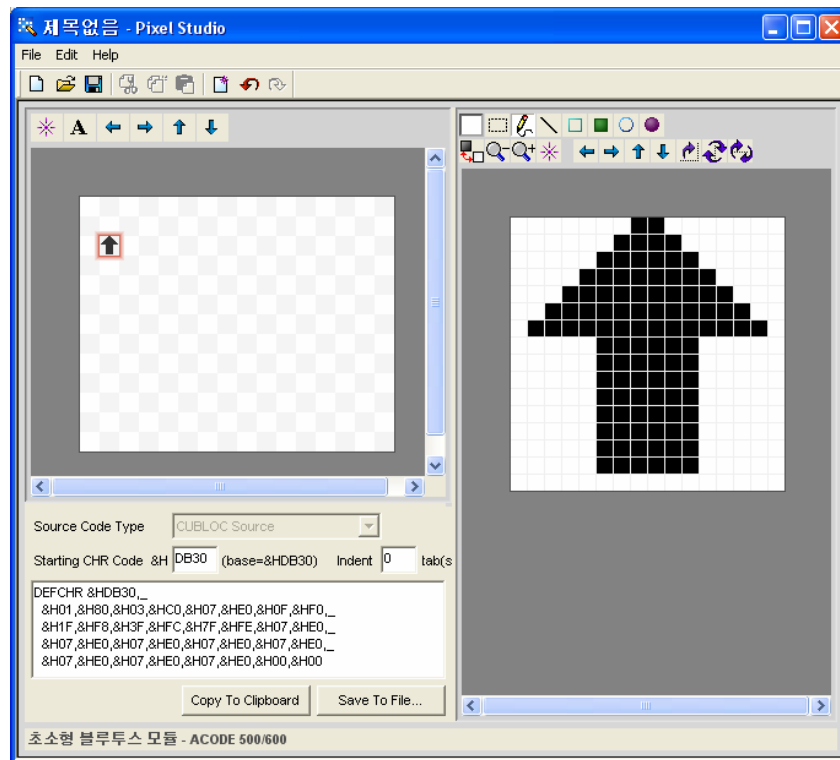
Parameters : code = Character Code(0DB30H to 0DBFFH)

data = 32 Byte Bitmap Data

Operation : You can create custom characters using this command. Each character can be 16 x 16 bitmap character. Simply send 32 bytes of bitmap data. A total of 207 custom characters can be stored onto the data memory of GHLCD.

Define Character is stored in the RAM of the GHLCD. The user must use this command to write custom characters to the RAM of the GHLCD every time it is power-cycled (turned off and turned on).

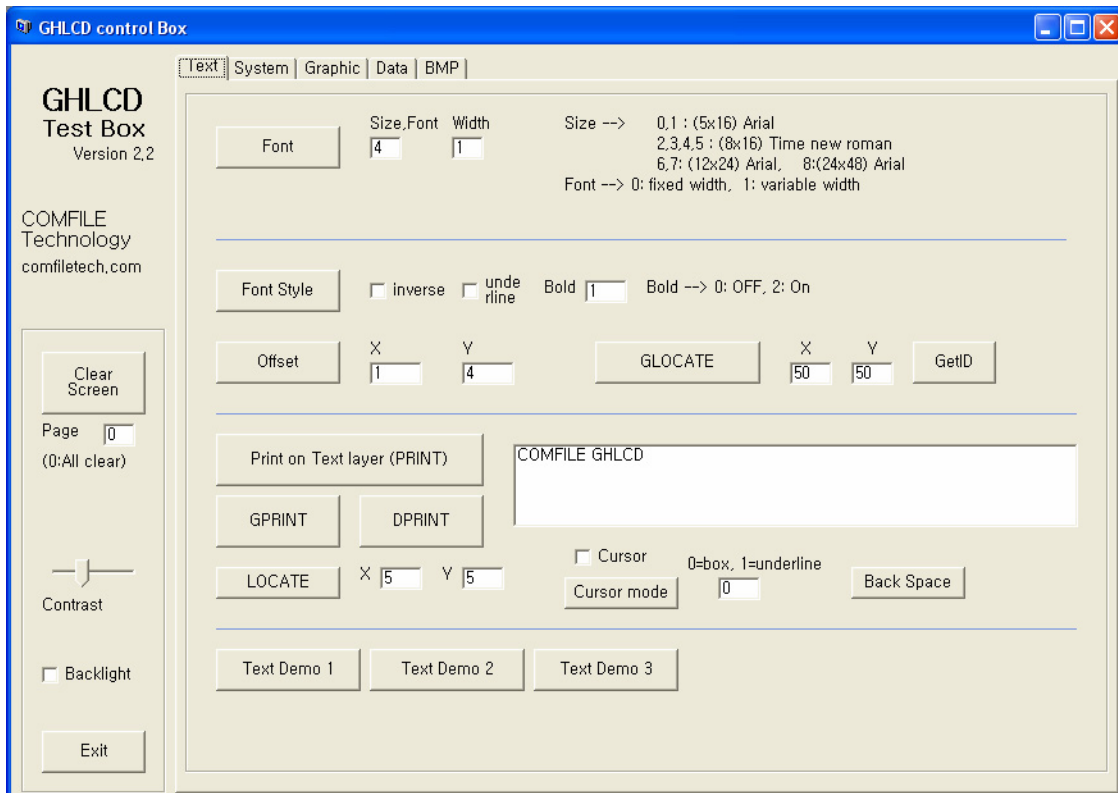
The user can easily create custom characters using the **Pixel Studio**, which can be downloaded on our website, www.comfiletech.com. The source code for Define Character will be generated for the user automatically.





GHLCD Control box

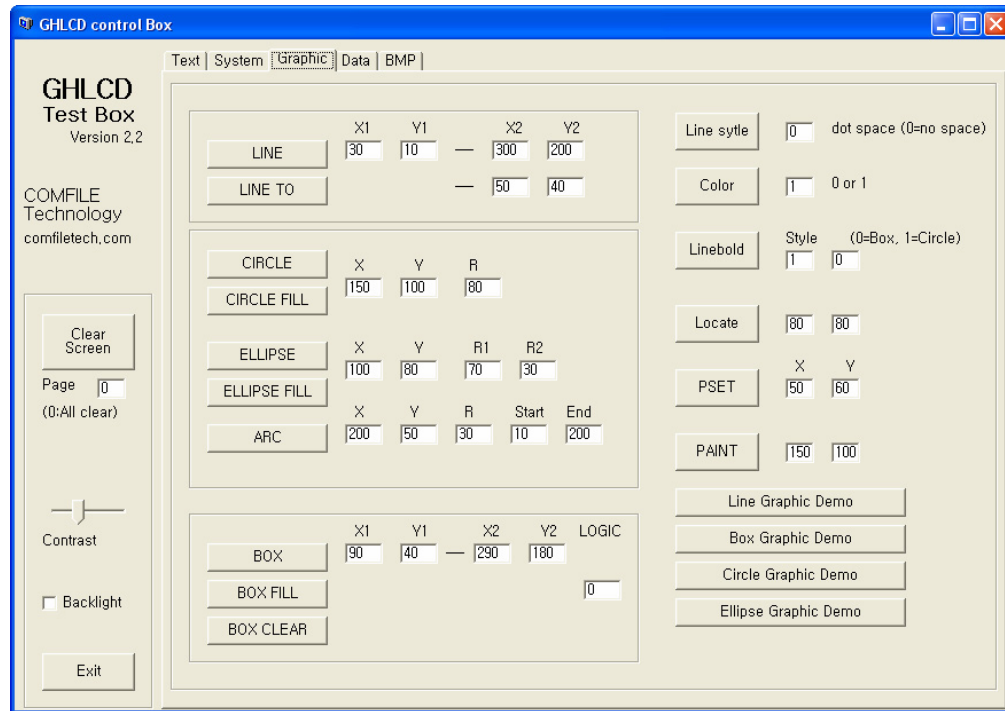
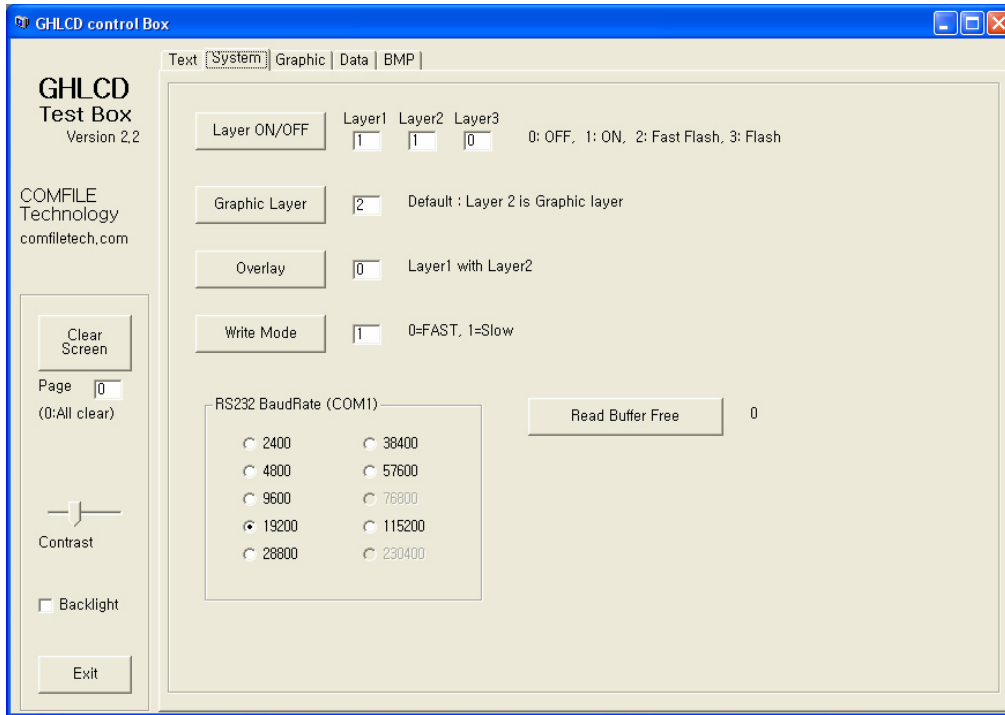
Use the **GHLCD Control Box** program to test your GHLCD when connected to a PC. (GHLCD Control Box can be downloaded on our website @ www.comfiletech.com.)



The user can test all features of GHLCD using this program.

1. First, please select the System tab and select the baudrate. The DIP switch on the back of the GHLCD can be maneuvered to set to a desired baudrate.
2. Text Layer can be tested by selecting the Text tab and writing text into the text box.
3. Please adjust the contrast bar on the left side of the Text tab if letters do not print clearly.

This program can be customized by the user to his/her own application by using our open-source Visual Basic code. The source code is included with the software. If you do not have Visual Basic installed on your PC, please copy all files in the Sys folder of the unzipped file to Windows/System32 directory on your computer before starting Testbox.exe

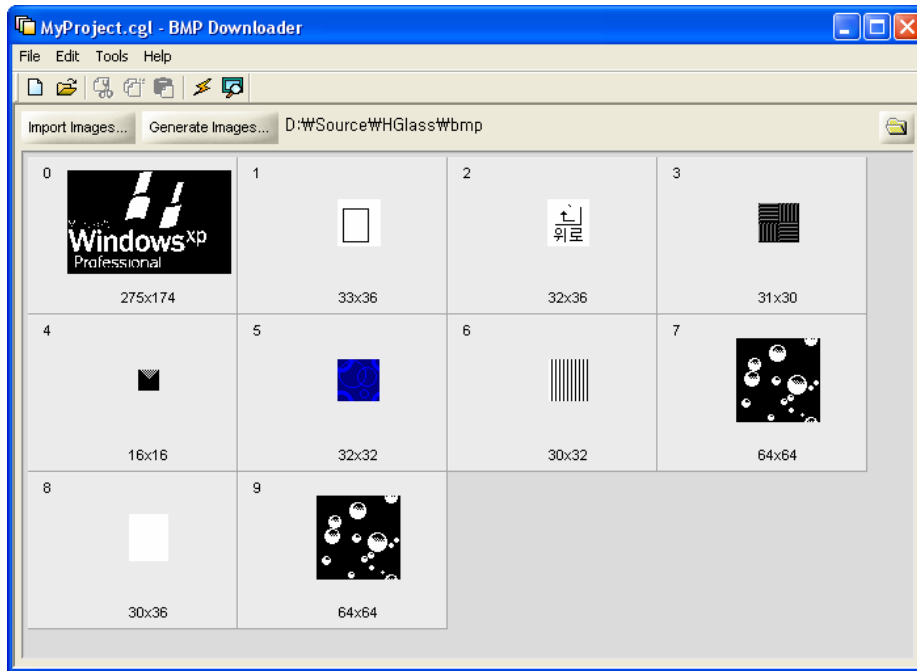




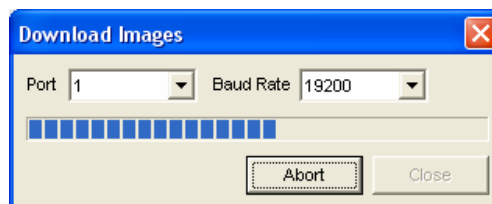
BMP Downloader

Use the **BMP Downloader** to download BMP files to the GHLCD. (Download @ www.comfiletech.com)

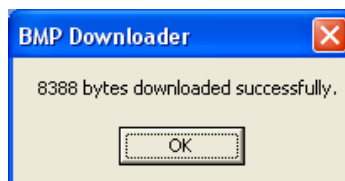
After creating your BMP file using any PC graphic editor, use the BMP Download to download to the Flash Memory of the GHLCD. Once downloaded, the BMP image will stay in the Flash Memory permanently until it is updated again.



1. Please press **Import Image**, then select the desired file to download. The BMP file must be in the **Monochrome bitmap**.
2. From **Tools**, please choose **Download image** and set the serial Port and the Baud rate.

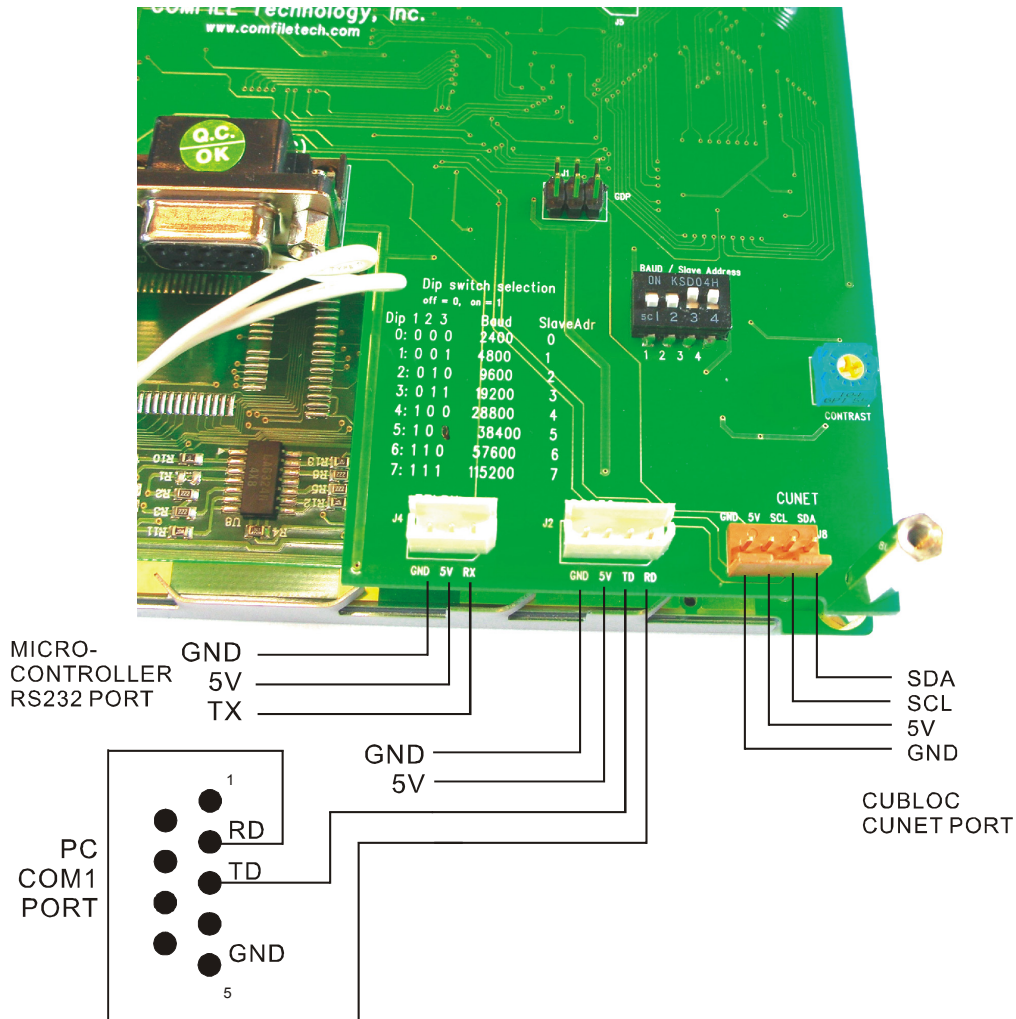


3. When download is finished, you will see the following message.





GHLCD connector



On the back of the GHB3224B, there are DIP switches to set the RS232 baud rate and I2C slave address. GHB3224 DIP Switch number 4 is not used.

DIP Switch	RS232 Baud Rate	I2C Slave Address
	2400	0
	4800	1

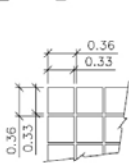
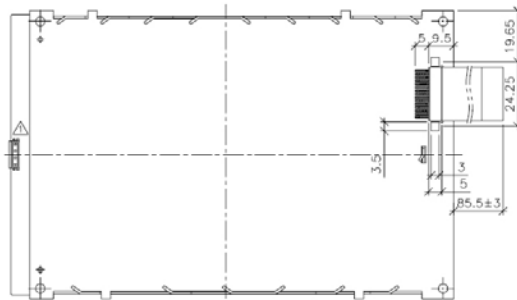
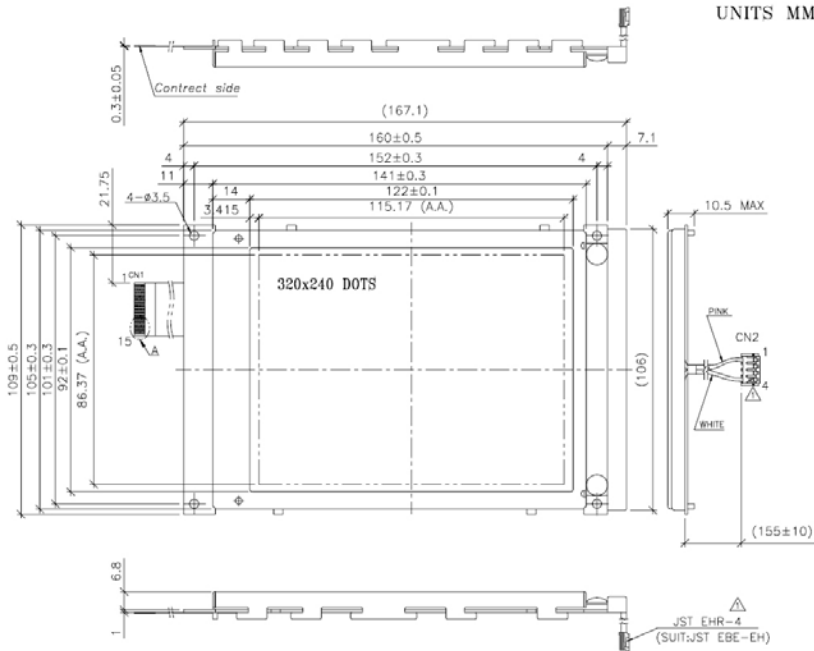


ON	9600	2
ON	19200	3
ON	28800	4
ON	38400	5
ON	57600	6
ON	115200	7

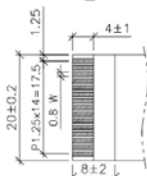
Please choose one communication method to use at a single time. (Either CuNET or RS232)



LCD dimension



DOT DETAIL S=30/1



DETAIL A
Scale:2/1

PIN ASSIGNMENT

CN1		CN2	
1. FG	8. D1	1. CFL HOT	
2. /DISP OFF	9. D2	2. NC	
3. FLM	10. D3	3. NC	
4. CL1	11. VSS	4. CFL GND	
5. CL2	12. VDD		
6. VSS	13. V0		
7. D0	14. VEE		
	15. VSS		

NOTE: TOLERANCE IS ±0.3 UNLESS OTHERWISE NOTED.

<End>