Color Graphic LCD with I2C Interface

GLCD-FLEXEL

User's Guide

2012

CONTENTS

1 Introduction.

“GLCD-FLEXEL” is a versatile, general-purpose Color Graphic LCD module with I2C interface. The module is designed as intelligent LCD with embedded 16-bit microcontroller and preloaded Graphic Library. The Graphic Library includes the graphic functions to print text and draw on the screen.
The embedded microcontroller provides the LCD control and takes care of communication with I2C bus. The module is connected to I2C bus as slave device and supports a simple command structure to communicate with Arduino or your microcontroller. Only two lines are used to connect the module to your controller.

“GLCD-FLEXEL” includes also a keypad port and supports the 4x4 matrix keypad or up to 8 buttons.

Arduino “GLCD-FLEXEL” Library is available. The library provides the complete set of functions to control the LCD module.

“GLCD-FLEXEL” can be used to replace the traditional character LCDs on Graphic LCD with 65K colors.

The module features:

- Communicate over standard I2C bus (100 Kbit speed) as slave device
- I2C signals are 5V tolerant, provides communication with 3.3V and 5V devices
- Use one power supply + 5V, includes 3.3V voltage regulator for LCD
- Graphic LCD with 65K colors and resolution 128 x 128 or 160 x 128
- Embedded 16-bit microcontroller with preloaded Graphic Library
- Graphic Library provides the complete set of functions to control LCD: draw line, circle, rectangle, bevel, arc, bar …
- Functions for screen color gradient rendering with 6 different styles
- Four preloaded fonts to print the small size text, medium size text, large size text and big digits
- Keypad Port to support a matrix keypad up to 16 keys (4 rows by 4 columns) or up to 8 buttons
- LCD backlight regulation with PWM control via software

2 Module connection.

2.1 I2C interface connector.

Table 2.1 shows the connector pin assignments.

Table 2.1

Pin No.	Pin Name	Description
1	SDA	I2C SDA signal. This pin should be connected directly to the SDA pin on your I2C Master device.
2	SCL	I2C SCL signal. This pin should be connected directly to the SCL pin on your I2C Master device.
3	GND	Ground connection
4	VDD	The +5V supply.

Note: GLCD-FLEXEL module includes pull-up resistors for SDA and SCL

2.2 Keypad connector.

The module supports a matrix keypad up to 16 keys (4 row x 4 column) or up to 8 buttons. Table 2.2 shows the connector pin assignments.

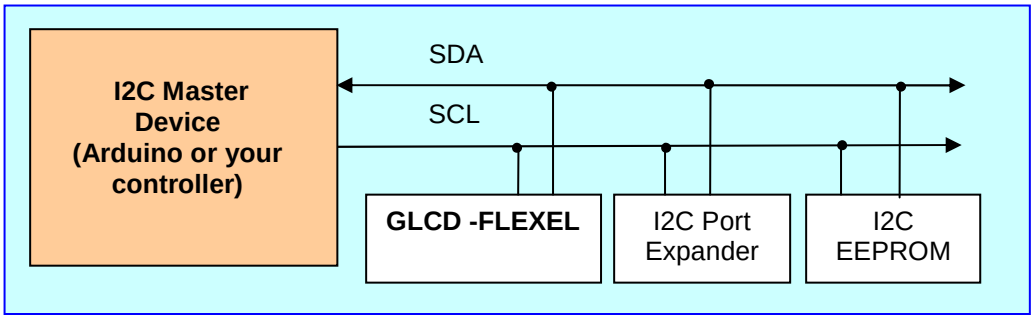
Table 2.2

Pin No.	Pin Name	Description
1	Row1	Keypad Row 1 / Button 1
2	Row2	Keypad Row 2 / Button 2
3	Row3	Keypad Row 3 / Button 3
4	Row4	Keypad Row 4 / Button 4
5	Col1	Keypad Col 1 / Button 5
6	Col2	Keypad Col 2 / Button 6
7	Col3	Keypad Col 3 / Button 7
8	Col4	Keypad Col 4 / Button 8
9	GND	Ground connection

2.3 Communication interface.

The I2C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device.
Note: GLCD-FLEXEL module includes the pull-up resistors for SDA and SCL lines.

The diagram below shows the module connections.



GLCD-FLEXEL is a Slave Divice on I2C bus and supports the I2C communication 100 kbps.

Each I2C device must have its own unique address (ID).
GLCD-FLEXEL address is 70 (0x46 HEX format).

The I2C Master Device initiates an I2C-bus data transfer through a series of commands.
To prevent the Master from handing the module due to an unfinished command sequence, the GLCD-FLEXEL module has a time-out feature. The delay between any two bytes of data coming from Master should be less than 255 ms. If this condition is not met, the module will time-out and clear the receive buffer. The module then starts to wait for the next command from the computer.

3 Commands.

The module is controlled using ASCII characters. The character decimal 254 (0xFE) is a command prefix. Any data sent to the GLCD-FLEXEL that is not prefixed by the command prefix (0xFE) will be displayed as chars on the LCD.

NOTE: To display the char on LCD, just send its ASCII number, 0x20 to 0x7F displays the standard set of characters. 0xFE is reserved for function command.

After power up GLCD-FLEXEL module set up the LCD and other peripherals. Initialization time is approximately 0.5 seconds. Arduino or your microcontroller software must have an appropriate delay before to send the command to GLCD-FLEXEL module.

Command Summary

Prefix	Command	Parameters	Description
			LCD Control Commands
0xFE	0x02	None	Init Graph
0xFE	0x03	None	Clear Screen
0xFE	0x04	2 bytes	Cursor Position
0xFE	0x05	None	Home
0xFE	0x06	None	Get Cursor X
0xFE	0x07	None	Get Cursor Y
0xFE	0x08	2 bytes	Set Color
0xFE	0x09	None	Get Color
0xFE	0x0A	1 byte	Font Type
0xFE	0x0B	1 byte	Font Orientation
0xFE	0x0C	4 bytes	Line
0xFE	0x0D	2 bytes	Line To
0xFE	0x0E	1 byte	Line Type
0xFE	0x0F	1 byte	Line Thickness
0xFE	0x10	4 bytes	Bar
0xFE	0x11	4 bytes	Rectangle
0xFE	0x12	3 bytes	Circle
0xFE	0x13	3 bytes	Fill Circle
0xFE	0x14	7 bytes	Arc
0xFE	0x15	5 bytes	Bevel
0xFE	0x16	5 bytes	Fill Bevel
0xFE	0x17	1 byte	Bevel Type
0xFE	0x18	10 bytes	Bar Gradient
0xFE	0x19	11 bytes	Bevel Gradient

0xFE	0x1A	2 bytes	Put Pixel
0xFE	0x1B	4 bytes	Draw Pixel
0xFE	0x1C	2 bytes	Push Color
			Module Control Commands
0xFE	0x30	None	Firmware Version
0xFE	0x31	1 byte	Set Keypad Mode
0xFE	0x32	None	Read Keypad
0xFE	0x33	None	Read Buttons
0xFE	0x34	1 byte	Set LCD Backlight

3.1 LCD Control Commands.

Init Graph

Syntax hexadecimal 0xFE 0x02]

Parameter	Length	Description
None	None	Set High Power pin

Description: I2C Master (Arduino or your microcontroller) issues the Init Graph command by sending two bytes 0xFE 0x02. This command makes the LCD hardware reset and initialization, fills LCD screen with BLACK color, sets a cursor position to upper left corner and sets the current color to WHITE.

This procedure runs also at module power up. Initializing time is approximately 0.5 seconds. Arduino or your microcontroller software must have an appropriate delay before to send the command to GLCD-FLEXEL module.

Clear Screen

Syntax hexadecimal 0xFE 0x03

Parameter	Length	Description
None	None	Clear LCD screen

Description: This command clears a LCD screen, fills the screen with a current color and moves a cursor to Home position.

Cursor Position

Syntax hexadecimal 0xFE 0x04 [x] [y]

Parameter	Length	Description
[x]	1 byte	X coordinate
[y]	1 byte	Y coordinate

Description: This command sets the cursor to position with x, y coordinates.

Home

Syntax hexadecimal 0xFE 0x05

Parameter	Length	Description
None	None	Set cursor to Home position

Description: This command sets the cursor to position with coordinates: x=0; y=0.

Get Cursor X

Syntax hexadecimal 0xFE 0x06

Parameter	Length	Description
None	None	Read the cursor position X

Description: I2C Master (Arduino) issues read the current cursor position X by sending two bytes 0xFE 0x06. Module returns 1 data byte with the current cursor position X.

Note: I2C interface Read Data command includes two transactions:
First transaction sends a command to read the data from slave device (data request);
Second transaction read the data from slave device.

Get Cursor Y

Syntax hexadecimal 0xFE 0x07

Parameter	Length	Description
None	None	Read the cursor position Y

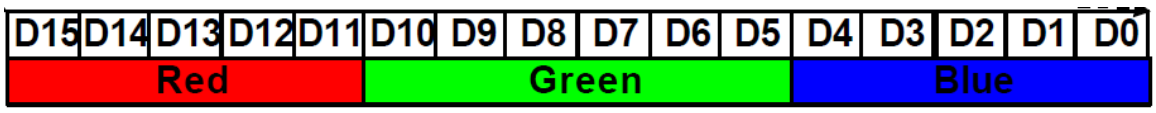
Description: I2C Master (Arduino) issues read the current cursor position Y by sending two bytes 0xFE 0x07. Module returns 1 data byte with the current cursor position Y.

Set Color

Syntax hexadecimal 0xFE 0x08 [colorByte1] [colorByte2]

Parameter	Length	Description
[colorByte1]	1 byte	Most significant data byte
[colorByte2]	1 byte	Least significant data byte

Description: This command sets the current drawing color. The module provides 16 bit color with RED, GREEN and BLUE combination in format: 5:6:5.



Color value examples:

Color	HEX Value
BLACK	0x0000
BLUE	0x001F
RED	0xF800
GREEN	0x07E0
CYAN	0x07FF
MAGENTA	0xF81F
YELLOW	0xFFE0
WHITE	0xFFFF

Get Color

Syntax hexadecimal 0xFE 0x09

Parameter	Length	Description
None	None	Read the current color

Description: I2C Master (Arduino) issues read the current color by sending two bytes 0xFE 0x09. Module returns 2 data bytes with 16 bit color value (first - most significant byte).

Font Type

Syntax hexadecimal 0xFE 0x0A [type]

Parameter	Length	Description
[type]	1 byte	Font type

Description: This command sets the current font type. The module includes four pre-loaded fonts: small size, medium size, large size and big digits.

Font Type	Type Number
SMALL	1
MEDIUM (default)	2
LARGE	3
BIG	4

Example: Command 0xFE 0x0A 0x03 sets a LARGE font type.

The SMALL, MEDIUM and LARGE fonts include ASCII characters from 0x20 (“Space”) to 0x7E (“~”).
The BIG DIGIT font includes ASCII characters from 0x20 (“Space”) to 0x3E (“>”).

Font Orientation

Syntax hexadecimal 0xFE 0x0B [orient]

Parameter	Length	Description
[orient]	1 bytes	Font orientation

Description: This command sets font orientation: horizontal (default) = 0 or vertical = 1.

Line

Syntax hexadecimal 0xFE 0x0C [x1] [y1] [x2] [y2]

Parameter	Length	Description
[x1]	1 byte	X coordinate of the start point
[y1]	1 byte	Y coordinate of the start point
[x2]	1 byte	X coordinate of the end point
[y2]	1 byte	Y coordinate of the end point

Description: This command draws a line with the current line type from the start point to the end point.
Example: Command 0xFE 0x0C 0x00 0x05 0x0A 0x14 draws a line from point (x=0; y=5) to point (x=10; y=20).

Line To

Syntax hexadecimal 0xFE 0x0D [x] [y]

Parameter	Length	Description
[x]	1 byte	End point X position
[y]	1 byte	End point Y position

Description: This command draws a line with the current line type from the current graphic cursor position to the given X, Y position. The graphic cursor position is moved to the end of the line.

Line Type

Syntax hexadecimal 0xFE 0x0E [type]

Parameter	Length	Description
[type]	1 byte	Line type

Description: This command sets a line type: SOLID (default) – 0; DOTTED – 1; DASHED-2

Example: Command 0xFE 0x0E 0x01 sets DOTTED line type.

Line Thickness

Syntax hexadecimal 0xFE 0x0F [thickness]

Parameter	Length	Description
[thickness]	1 byte	Line thickness

Description: This command sets a line thickness: NORMAL (default, thickness is 1 pixel) – 0; THICK (thicknees is 3 pixels) – 1.

Example: Command 0xFE 0x0F 0x01 sets THICK line.

Bar

Syntax hexadecimal 0xFE 0x10 [left] [top] [right] [bottom]

Parameter	Length	Description
[left]	1 byte	X position of the left top corner
[top]	1 byte	Y position of the left top corner
[right]	1 byte	X position of the right bottom corner
[bottom]	1 byte	Y position of the right bottom corner

Description: This command draws a bar given the left, top and right, bottom corners with the current color.

Rectangle

Syntax hexadecimal 0xFE 0x11 [left] [top] [right] [bottom]

Parameter	Length	Description
[left]	1 bytes	X position of the left top corner
[top]	1 byte	Y position of the left top corner
[right]	1 byte	X position of the right bottom corner
[bottom]	1 byte	Y position of the right bottom corner

Description: This command draws a rectangle given the left, top and right, bottom corners with the current color. Current line type is used.

Circle

Syntax hexadecimal 0xFE 0x12 [x] [y] [radius]

Parameter	Length	Description
[x]	1 byte	Center X position
[yp]	1 byte	Center Y position
[radius]	1 byte	The radius of the circle

Description: This command draws a circle with the given center and radius. Current color is used.

Fill Circle

Syntax hexadecimal 0xFE 0x13 [x] [y] [radius]

Parameter	Length	Description
[x]	1 bytes	Center X position
[yp]	1 byte	Center Y position
[radius]	1 byte	The radius of the circle

Description: This command draws a filled circle with the given center and radius. Current color is used.

Arc

Syntax hexadecimal 0xFE 0x14 [xL] [yT] [xR] [yB] [radius1] [radius2] [octant]

Parameter	Length	Description
[xL]	1 byte	X location of the upper left center
[yT]	1 byte	Y location of the upper left center
[xR]	1 byte	X location of the lower right center
[yB]	1 byte	Y location of the lower right center
[radius1]	1 byte	The smaller radius of the two concentric circles that defines the thickness of the object
[radius2]	1 byte	The larger radius of the two concentric circles that defines the thickness of the object
[octant]	1 byte	Bitmask of the octant that will be drawn. Moving in a clockwise direction from x=0, y= +radius • Bit0 : first octant • Bit1 : second octant • Bit2 : third octant • Bit3 : fourth octant • Bit4 : fifth octant • Bit5 : sixth octant • Bit6 : seventh octant • Bit7 : eight octant **Examples:** Set [octant]=0x06 to draw second and third octant. Set [octant] = 0xFF to draw a full arc.

Description: This command draws a filled circle with the given center and radius. Current color is used.

Bevel

Syntax hexadecimal 0xFE 0x15 [x1] [y1] [x2] [y2] [radius]

Parameter	Length	Description
[x1]	1 byte	X coordinate of the upper left center of the circle that draws the rounded corners.
[y1]	1 byte	Y coordinate of the upper left center of the circle that draws the rounded corners.
[x2]	1 byte	X coordinate of the lower right center of the circle that draws the rounded corners.
[y2]	1 byte	Y coordinate of the lower right center of the circle that draws the rounded corners.
[radius]	1 byte	Defines the radius of the circle, that draws the corners.

Description: This command draws a beveled figure on the screen. For a pure circular object x x1 = x2 and y1 = y2. For a rectangular object radius = 0.

Fill Bevel

Syntax hexadecimal 0xFE 0x16 [x1] [y1] [x2] [y2] [radius]

Parameter	Length	Description
[x1]	1 byte	X coordinate of the upper left center of the circle that draws the rounded corners.
[y1]	1 byte	Y coordinate of the upper left center of the circle that draws the rounded corners.
[x2]	1 byte	X coordinate of the lower right center of the circle that draws the rounded corners.
[y2]	1 byte	Y coordinate of the lower right center of the circle that draws the rounded corners.
[radius]	1 byte	Defines the radius of the circle that draws the rounded corners.

Description: This command draws a filled beveled figure on the screen. For a pure circular object x x1 = x2 and y1 = y2. For a rectangular object radius = 0.

Bevel Type

Syntax hexadecimal 0xFE 0x17 [type]

Parameter	Length	Description
[type]	1 byte	Bevel type is set using the following: • DRAWFULLBEVEL = 0 to draw the full shape • DRAWTOPBEVEL = 1 to draw the upper half portion • DRAWBOTTOMBEVEL = 2 to draw the lower half portion

Description: This command sets the fill bevel type to be drawn

Example: Command 0xFE 0x17 0x01 sets bevel type to draw the upper half portion.

Bar Gradient

Syntax hexadecimal 0xFE 0x18 [left] [top] [right] [bottom] [MSBcolor1] [LSBcolor1] [MSBcolor2] [LSBcolor2] [length] [direction]

Parameter	Length	Description
[left]	1 byte	X position of the left top corner
[top]	1 byte	Y position of the left top corner
[right]	1 byte	X position of the right bottom corner
[bottom]	1 byte	Y position of the right bottom corner
[MSBcolor1]	1 byte	Start color for the gradient (most significant byte)
[LSBcolor1]	1 byte	Start color for the gradient (least significant byte)
[MSBcolor2]	1 byte	End color for the gradient (most significant byte)
[LSBcolor2]	1 byte	End color for the gradient (least significant byte)
[length]	1 byte	From 0-100%. How much of gradient is wanted
		Gradient Direction: • GRAD_DOWN = 1 gradient changes in vertical direction • GRAD_RIGHT = 2 gradient changes in the

[direction]	1 byte	horizontal direction • GRAD_UP = 3 gradient changes in vertical direction • GRAD_LEFT = 4 gradient changes in the horizontal direction • GRAD_DOUBLE_VER = 5 two gradient transitions in the vertical direction • GRAD_DOUBLE_HOR = 6 two gradient transitions in the gorizontal direction

Description: This command draws the bar gradient given the left, top and right, bottom corners.

Bevel Gradient

Syntax hexadecimal 0xFE 0x19 [x1] [y1] [x2] [y2] [radius] [MSBcolor1] [LSBcolor1] [MSBcolor2] [LSBcolor2] [length] [direction]

Parameter	Length	Description
[x1]	1 byte	X position of the upper left center of the circle that draws the rounded corners
[y1]	1 byte	Y position of the upper left center of the circle that draws the rounded corners
[x2]	1 byte	X position of the lower right center of the circle that draws the rounded corners
[y2]	1 byte	X position of the lower right center of the circle that draws the rounded corners
[radius]	1 byte	Defines the radius of the circle that draws the rounded corners
[MSBcolor1]	1 byte	Start color for the gradient (most significant byte)
[LSBcolor1]	1 byte	Start color for the gradient (least significant byte)
[MSBcolor2]	1 byte	End color for the gradient (most significant byte)
[LSBcolor2]	1 byte	End color for the gradient (least significant byte)
[length]	1 byte	From 0-100%. How much of gradient is wanted
[direction]	1 byte	Gradient Direction: • GRAD_DOWN = 1 gradient changes in vertical direction • GRAD_RIGHT = 2 gradient changes in the horizontal direction • GRAD_UP = 3 gradient changes in vertical direction • GRAD_LEFT = 4 gradient changes in the horizontal direction • GRAD_DOUBLE_VER = 5 two gradient transitions in the vertical direction • GRAD_DOUBLE_HOR = 6 two gradient transitions in the gorizontal direction

Description: This renders a gradient on the screen. It works the same as the Fill Bevel command, except a gradient out of color1 and color2 is drawn depending on the direction.

Put Pixel

Syntax hexadecimal 0xFE 0x1A [x] [y]

Parameter	Length	Description
[x]	1 byte	Pixel X position
[y]	1 byte	Pixel Y position

Description: This command fills a pixel with the coordinates (x, y) with the current color.

Draw Pixel

Syntax hexadecimal 0xFE 0x1B [x] [y] [colorByte1] [colorByte2]

Parameter	Length	Description
[x]	1 byte	Pixel X position
[y]	1 byte	Pixel Y position
[colorByte1]	1 byte	Most significant color byte
[colorByte2]	1 byte	Least significant color byte

Description: This command fills a pixel with the coordinates (x, y) with the 16 bit color.

Push Color

Syntax hexadecimal 0xFE 0x1C [colorByte1] [colorByte2]

Parameter	Length	Description
[colorByte1]	1 byte	Most significant color byte
[colorByte2]	1 byte	Least significant color byte

Description: This command fills a pixel with the current graphic cursor position with the 16 bit color.

3.2 Module Control Commands.

Read Firmware Version

Syntax hexadecimal 0xFE 0x30

Parameter	Length	Description
None	None	Read the firmware version number

Description: I2C Master (Arduino) issues the read firmware version command by sending two bytes 0xFE 0x30. Module returns data byte with firmware version number.

Set Keypad Mode

Syntax hexadecimal 0xFE 0x31 [mode]

Parameter	Length	Description
[mode]	1 byte	Set keypad mode

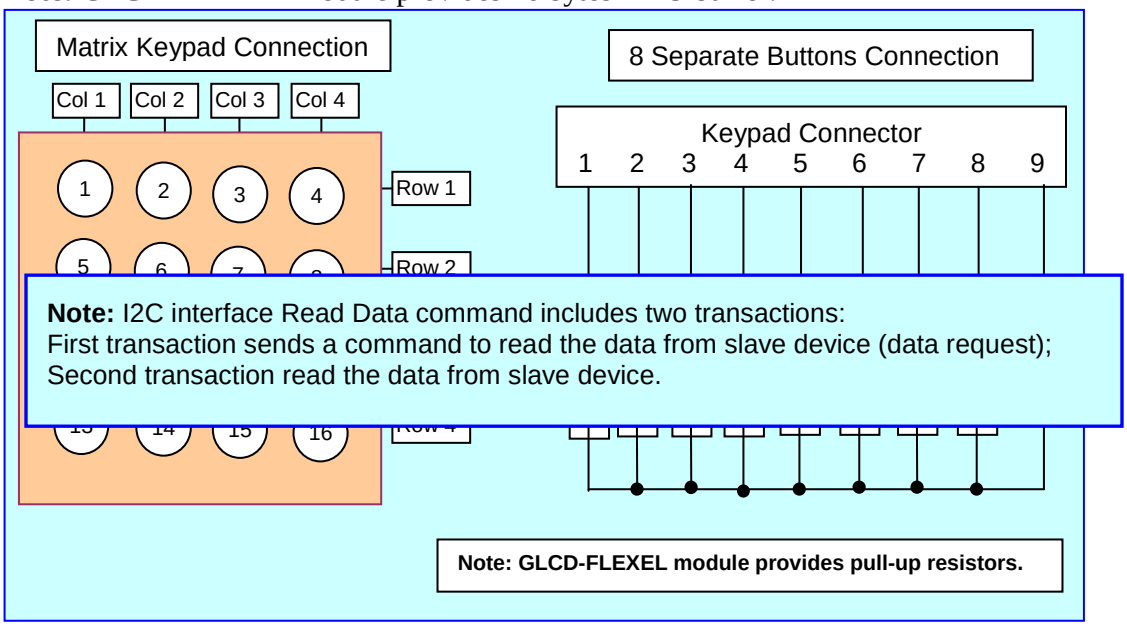
Description: I2C Master (Arduino) sets the keypad mode by sending two bytes 0xFE 0x31 and 1 data byte. Data byte defines the keypad mode.
Data byte = 0x00 - matrix keypad 4 row x 4 column (default at power up).
Data byte = 0x01 - 8 button port.

Read Keypad

Syntax hexadecimal 0xFE 0x32

Parameter	Length	Description
None	None	Read the keypad port buffer

Description: I2C Master (Arduino) issues read the keypad by sending two bytes 0xFE 0x32. Module returns 1 data bytes. If there is no key data in the keypad buffer, 0 is returned. If there is key data, a key code from 0x01 to 0x10 is returned, depending on what key was pressed.
Note: GLCD-FLEXEL module provides 16 bytes FIFO buffer.



Read Buttons
Syntax hexadecimal 0xFE 0x33

Parameter	Length	Description
None	None	Read the keypad port buffer

Description: I2C Master (Arduino) issues read the keypad port by sending two bytes 0xFE 0x33. Module returns 1 data bytes. If there is no data in the keypad buffer, 0 is returned. If there is data, a button code from 0x01 to 0x08 is returned, depending on what button was pressed.
Note: GLCD-FLEXEL module provides 16 bytes FIFO buffer.

Set LCD Backlight
Syntax hexadecimal 0xFE 0x34 [data]

Parameter	Length	Description
[data]	1 byte	Set LCD backlight

Description: GLCD-FLEXEL module provides the backlight regulation with PWM control. I2C Master (Arduino or your microcontroller) sets the LCD backlight by sending two bytes 0xFE 0x31 and 1 data byte. Data byte defines the backlight value:
Data byte = 0x00 - backlight off.
Data byte = 0xFF - max backlight (default at power up).

4 Arduino “GLCD-FLEXEL” Library.

The library provides the complete set of functions to control the LCD module and gives you an opportunity to spend more time on your project, not on LCD “pixel level” control.
The library functions include the I2C interface commands and take care of Arduino board communication with “GLCD-FLEXEL”.

You can download the latest version of the library from website.

Arduino Library Function Summary

The library includes GLCDI2C class with the next public functions:

Function Name	Description
print	Print text to the LCD. Syntax: lcd.print(data) lcd.print(data, BASE) Parameters: data – the data to print (char, byte, int, long, or string) BASE (optional): the base in wihich to print numbers: BIN for binary, DEC for decimal, OCT for octal, HEX for hexadecimal. Example output “Hello, World!”: lcd.print(“Hello, World!”); Example print integer value 25 on screen: Tmp=50; lcd.print(Tmp/2);
init	Initialize LCD. void init(void)
clear	Clear the screen. void clear(void)
cursor	Set the graphic cursor position. void cursor(unsigned int x, unsigned int y)
home	Set the cursor to Home position (x = 0; y = 0). void home(void)
setColor	Set the current color. void setColor(unsigned int color)
getColor	Read the current color. unsigned int getColor(void)
fontType	Set the font type. void fontType(unsigned char font)
fontOrientation	Set the font orientation (horizontal or vertical). void fontOrientation(unsigned char font_orient)
line	Draw the line. void line(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2)
lineTo	Draw the line from the cursor position to new point. void lineTo(unsigned int x, unsigned int y)
lineType	Set the line type. void lineType(unsigned char line_type)
lineSize	Set the line size. void lineSize(unsigned char line_size)
bar	Draw the bar. void bar(unsigned int left, unsigned int top, unsigned int right, unsigned int bottom)
rectangle	Draw the rectangle. void rectangle(unsigned int left, unsigned int top, unsigned int right, unsigned int bottom)
circle	Draw the circle. void circle(unsigned int x, unsigned int y, unsigned int radius)
fillCircle	Draw the filled circle.

	void fillCircle(unsigned int x, unsigned int y, unsigned int radius)
arc	Draw the arc. void arc(unsigned int xL, unsigned int yT, unsigned int xR, unsigned int yB, unsigned int r1, unsigned int r2, unsigned char octant)
bevel	Draw the bevel. void bevel(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2, unsigned int radius)
fillBevel	Draw the filled bevel. void fillBevel(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2, unsigned int radius)
bevelType	Set the bevel type. void bevelType(unsigned char type)
BarGradient	Draw the bar gradient. void barGradient(unsigned int left, unsigned int top, unsigned int right, unsigned int bottom, unsigned int color1, unsigned int color2, unsigned int length, unsigned char direction)
bevelGradient	Draw the bevel gradient. void bevelGradient(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2, unsigned int rad, unsigned int color1, unsigned int color2, unsigned int length, unsigned char direction)
putPixel	Fill a pixel with the coordinates (x, y) with the current color. void putPixel(unsigned int x, unsigned int y)
drawPixel	Fill a pixel with the coordinates (x, y) with the 16 bit color. void drawPixel(unsigned int x, unsigned int y, unsigned int color)
pushColor	Fill a pixel with the current graphic cursor position with the 16 bit color. void pushColor(unsigned int color)
write	Send a data byte to LCD. virtual size_t write(unsigned char)
firmware	Read a LCD firmware version. unsigned int firmware(void)
keypadMode	Set the Keypad Port mode: matrix keypad or buttons. void keypadMode(unsigned char mode)
keypad	Read a keypad buffer from LCD unsigned int keypad (void)
button	Read a button buffer from LCD unsigned int button (void)
backlight	Set the LCD backlight. void backlight(unsigned char light)