LCDWIKI GUI lib Manual

1.Introduction

The LCDWIKI GUI lib is the core graphics library for all our displays, providing a common set of graphics primitives (points, lines, circles, etc.).   It needs to be paired with a hardware-specific library for each display device we carry (to handle the lower-level functions).

The LCDWIKI GUI lib have the base class.so all functions in this lib should be called by the subclass.

2.FUNCTIONS DECLARATION

definiens	LCDWIKI_GUI(void)
function	The main class constructor when using 8bit or 16bit or spi display modules.
parameters	None
returned value	None
notes	None

definiens	virtual uint16_t Color_To_565(uint8_t r, uint8_t g, uint8_t b)
function	Pass three 8bits colour value and get the 16bits colour value
parameters	r : the 8bits red value g : the 8bits green value b : the 8bits blue value
returned value	The 16bits colour value(rrrrrggggggbbbbb)
notes	This is virtual function and it is defined by the subclass

definiens	virtual void Draw_Pixe(int16_t x, int16_t y, uint16_t color)
function	Using color value to draw a single point
parameters	x : the x coordinate of the piexl y : the y coordinate of the piexl color : the color value of the piexl
returned value	None
notes	This is virtual function and it is defined by the subclass

definiens	virtual void Fill_Rect(int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
function	Using color value to draw a filled rectangle with w width and h height in x and y coordinate
parameters	x : the x coordinate of the start-corner y : the y coordinate of the start-corner w : the width of the rectangle h : the height of the rectangle color : the color value of the filled rectangle
returned value	None
notes	This is virtual function and it is defined by the subclass

definiens	virtual void Set_Addr_Window(int16_t x1, int16_t y1, int16_t x2, int16_t y2)
function	Set display area bewteen two point
parameters	x1 : the x coordinate of the start-corner y1 : the y coordinate of the start-corner x2 : the x coordinate of the end-corner y2 : the y coordinate of the end-corner
returned value	None
notes	This is virtual function and it is defined by the subclass

definiens	virtual void Push_Any_Color(uint16_t * block, int16_t n, bool first, uint8_t flags)
function	Set a large number of color values at a time
parameters	block : the array of colour values n : the number of colour values first : 1- First set the command of write color value 0-have set the command of write color value flags : 0-read color value from RAM 1-read color value from flash
returned value	None
notes	This is virtual function and it is defined by the subclass

definiens	virtual int16_t Read_GRAM(int16_t x, int16_t y, uint16_t *block, int16_t w, int16_t h)
function	Read colour value from GRAM
parameters	x : the x coordinate of the start-corner y : the y coordinate of the start-corner block : the array of saving colour value w : the width of the Read area h : the heigth of the Read area
returned value	0-successful
notes	This is virtual function and it is defined by the subclass

definiens	virtual int16_t Get_Height(void) const
function	Get the display height
parameters	None
returned value	The diaplay height
notes	This is virtual function and it is defined by the subclass

definiens	virtual int16_t Get_Width(void) const
function	Get the display width
parameters	None
returned value	The diaplay width
notes	This is virtual function and it is defined by the subclass

definiens	void Set_Draw_color(uint16_t color)
function	Set the drawing color
parameters	color : the 16bits Drawing color value
returned value	None
notes	None

definiens	void Set_Draw_color(uint16_t color)
function	Set the drawing color
parameters	color : the 16bits Drawing color value
returned value	None

notes	None

definiens	void Set_Draw_color(uint8_t r, uint8_t g, uint8_t b)
function	Set the drawing color
parameters	r : the 8bits red value g : the 8bits green value b : the 8bits blue value
returned value	None
notes	None

definiens	uint16_t Get_Draw_color(void) const
function	get the drawing color
parameters	None
returned value	The 16bits drawing color value(rrrrrggggggbbbbb)
notes	None

definiens	void Draw_Pixel(int16_t x, int16_t y)
function	Draw a single point
parameters	x : the x coordinate of the piexl y : the y coordinate of the piexl
returned value	None
notes	None

definiens	uint16_t Read_Pixel(int16_t x, int16_t y)
function	Get the color value of a single point
parameters	x : the x coordinate of the piexl y : the y coordinate of the piexl
returned value	16bits color value of a single point
notes	The mould must be readable,this functon can be used normally.

definiens	void Draw_Fast_VLine(int16_t x, int16_t y, int16_t h)
function	Quickly draw out a vertical line
parameters	x : the x coordinate of the start point y : the y coordinate of the start point h : the height of the line
returned value	None
notes	None

definiens	void Draw_Fast_HLine(int16_t x, int16_t y, int16_t w)
function	Quickly draw out a horizontal line
parameters	x : the x coordinate of the start point y : the y coordinate of the start point h : the width of the line
returned value	None
notes	None

definiens	void Fill_Screen(uint16_t color)
function	Fill whole screen area
parameters	color : 16bits color value
returned value	None
notes	None

definiens	void Fill_Screen(uint8_t r, uint8_t g, uint8_t b)
function	Fill whole screen area
parameters	r : the 8bits red value g : the 8bits green value b : the 8bits blue value
returned value	None
notes	The r,g,b is converted to 16bits value(rrrrrggggggbbbbb)

definiens	void Draw_Line(int16_t x1, int16_t y1, int16_t x2, int16_t y2)
function	Draw a line be
parameters	r : the 8bits red value g : the 8bits green value b : the 8bits blue value
returned value	None
notes	The r,g,b is converted to 16bits value(rrrrrggggggbbbbb)

definiens	void Draw_Rectangle(int16_t x1, int16_t y1, int16_t x2, int16_t y2)
function	Draw a rectangle between two points
parameters	x1 : the x coordinate of the start point y1 : the y coordinate of the start point x2 : the x coordinate of the end point y2 : the y coordinate of the end point
returned value	None
notes	None

definiens	void Fill_Rectangle(int16_t x1, int16_t y1, int16_t x2, int16_t y2)
function	Draw a filled rectangle between two points
parameters	x1 : the x coordinate of the start point y1 : the y coordinate of the start point x2 : the x coordinate of the end point y2 : the y coordinate of the end point
returned value	None
notes	None

definiens	void Draw_Round_Rectangle(int16_t x1, int16_t y1, int16_t x2, int16_t y2, uint8_t radius)
function	Draw a rectangle with slightly rounded corners between two points

parameters	x1 : the x coordinate of the start point y1 : the y coordinate of the start point x2 : the x coordinate of the end point y2 : the y coordinate of the end point radius : radius of the rounded corners, the minimum value is 5
returned value	None
notes	If the radius is smaller than the minimum value ,the round rectangle will not be drawn

definiens	void Fill_Round_Rectangle(int16_t x1, int16_t y1, int16_t x2,int16_t y2, int16_t radius)
function	Draw a filled rectangle with slightly rounded corners between two points
parameters	x1 : the x coordinate of the start point y1 : the y coordinate of the start point x2 : the x coordinate of the end point y2 : the y coordinate of the end point radius : radius of the rounded corners, the minimum value is 5
returned value	None
notes	If the radius is smaller than the minimum value ,the round rectangle will not be drawn

definiens	void Draw_Circle(int16_t x, int16_t y, int16_t radius)
function	Draw a circle with a specified radius
parameters	x : the x coordinate of the center of the circle y : the y coordinate of the center of the circle

	radius : radius of the circle
returned value	None
notes	None

definiens	void Draw_Circle_Helper(int16_t x0, int16_t y0, int16_t radius, uint8_t cornername)
function	Draw rounded corners with a specified radius
parameters	x0 : the x coordinate of the center of the rounded corner y0 : the y coordinate of the center of the rounded corner radius : radius of the rounded corners, the minimum value is 5 cornername : the order number of the rounded corners, 1 : top left corner 2 : top right corner 4 : lower right corner 8 : lower left quarter
returned value	None
notes	If the radius is smaller than the minimum value ,the round rectangle will not be drawn.you can draw several rounded corners at a time

definiens	void Fill_Circle(int16_t x, int16_t y, int16_t radius)
function	Draw a filled circle with a specified radius
parameters	x : the x coordinate of the center of the circle y : the y coordinate of the center of the circle radius : radius of the circle

returned value	None
notes	None

definiens	void Fill_Circle_Helper(int16_t x0, int16_t y0, int16_t r, uint8_t cornername,int16_t delta)
function	Draw filled rounded corners with a specified radius
parameters	x0 : the x coordinate of the center of the rounded corner y0 : the y coordinate of the center of the rounded corner r : radius of the rounded corner cornername : the order number of the rounded corners 1 : right rounded corner 2 : left rounded corner delta : Non - circular area height
returned value	None
notes	you can draw filled several rounded corners at a time

definiens	void Draw_Triangle(int16_t x0, int16_t y0, int16_t x1, int16_t y1,int16_t x2, int16_t y2)
function	Draw a triangle between three points
parameters	x0 : the x coordinate of the start point of the triangle bottom y0 : the y coordinate of the start point of the triangle bottom x1 : the x coordinate of the triangular vertex y1 : the x coordinate of the triangular vertex x2 : the x coordinate of the end point of the triangle bottom y2 : the y coordinate of the end point of the triangle bottom

returned value	None
notes	None

definiens	void Fill_Triangle(int16_t x0, int16_t y0, int16_t x1, int16_t y1,int16_t x2, int16_t y2)
function	Draw a filled triangle between three points
parameters	x0 : the x coordinate of the start point of the triangle bottom y0 : the y coordinate of the start point of the triangle bottom x1 : the x coordinate of the triangular vertex y1 : the y coordinate of the triangular vertex x2 : the x coordinate of the end point of the triangle bottom y2 : the y coordinate of the end point of the triangle bottom
returned value	None
notes	None

definiens	void Draw_Bit_Map(int16_t x, int16_t y, int16_t sx, int16_t sy, const uint16_t *data, int16_t scale)
function	Draw a bitmap on the screen
parameters	x : the x coordinate of the top left corner of bitmap y : the y coordinate of the top left corner of bitmap sx : width of the bitmap sy : height of the bitmap data : array containing the bitmap-data scale : scaling factor. Each pixel in the bitmap will be drawn as <scale>x<scale> pixels on screen

returned value	None
notes	None

definiens	void Set_Text_Cousur(int16_t x, int16_t y)
function	Set text position in screen
parameters	x : the x coordinate of the text y : the y coordinate of the text
returned value	None
notes	None

definiens	int16_t Get_Text_X_Cousur(void) const
function	get the x coordinate of the text
parameters	None
returned value	the x coordinate of the text
notes	None

definiens	int16_t Get_Text_Y_Cousur(void) const
function	get the y coordinate of the text
parameters	None
returned value	the y coordinate of the text
notes	None

definiens	void Set_Text_colour(uint16_t color)
function	Set the text color value
parameters	Color : the 16bits color value of the text
returned value	None
notes	None

definiens	void Set_Text_colour(uint8_t r, uint8_t g, uint8_t b)
function	Set the text color value
parameters	r : the 8bits red value of the text g : the 8bits green value of the text b : the 8bits blue value of the text
returned value	None
notes	None

definiens	uint16_t Get_Text_colour(void) const
function	get the text color value
parameters	None
returned value	the 16bits color value of the text
notes	None

definiens	void Set_Text_Back_colour(uint16_t color)
function	set the background color value of the text

parameters	Color : the 16bits background color value of the text
returned value	None
notes	None

definiens	void Set_Text_Back_colour(uint8_t r, uint8_t g, uint8_t b)
function	set the background color value of the text
parameters	r : the 8bits red value of the text g : the 8bits green value of the text b : the 8bits blue value of the text
returned value	None
notes	None

definiens	void Set_Text_Back_colour(uint8_t r, uint8_t g, uint8_t b)
function	set the background color value of the text
parameters	r : the 8bits red value of the text g : the 8bits green value of the text b : the 8bits blue value of the text
returned value	None
notes	None

definiens	uint16_t Get_Text_Back_colour(void) const
function	get the background color value of the text
parameters	None

returned value	the 16bits background color value of the text
notes	None

definiens	void Set_Text_Size(uint8_t s)
function	set the size of the text
parameters	s : the size of the text
returned value	None
notes	None

definiens	uint8_t Get_Text_Size(void) const
function	Get the size of the text
parameters	None
returned value	The size value of the text
notes	None

definiens	void Set_Text_Mode(boolean mode)
function	Set overlap mode of the text
parameters	mode : 0-no overlap 1-overlap
returned value	None
notes	If the mode is overlap,the background color setting of the text is invalid.

definiens	boolean Get_Text_Mode(void) const
function	get the overlap mode value of the text
parameters	None
returned value	0-no overlap 1-overlap
notes	If the mode is overlap,the background color setting of the text is invalid.

definiens	size_t Print(uint8_t *st, int16_t x, int16_t y)
function	Print a string at the specified coordinates
parameters	st : the string to print x : the x coordinate of the top left corner of the first character y : the y coordinate of the top left corner of the first character
returned value	The Number of characters
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Print_String(const uint8_t *st, int16_t x, int16_t y)
function	Print a constant string at the specified coordinates
parameters	st : the constant string to print x : the x coordinate of the top left corner of the first character y : the y coordinate of the top left corner of the first character
returned value	None
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Print_String(uint8_t *st, int16_t x, int16_t y)
function	Print a string at the specified coordinates
parameters	st : the string to print x : the x coordinate of the top left corner of the first character y : the y coordinate of the top left corner of the first character
returned value	None
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Print_String(String st, int16_t x, int16_t y)
function	Using string class to print a string at the specified coordinates
parameters	st : the string object x : the x coordinate of the top left corner of the first character y : the y coordinate of the top left corner of the first character
returned value	None
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Print_Number_Int(long num, int16_t x, int16_t y, int16_t length, uint8_t filler, int16_t system)
function	Print a specified length Integral number at the specified coordinates
parameters	num : the value to print (-2,147,483,648 to 2,147,483,647) INTEGERS ONLY

	x : the x coordinate of the top left corner of the first number/sign y : the y coordinate of the top left corner of the first number/sign length : minimum number of digits/characters (including sign) to display filler : filler character to use to get the minimum length. The character will be inserted in front of the number, but after the sign. Default is ' ' (space). system : 8-octal 10- decimal 16- hexadecimal
returned value	None
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Print_Number_Float(double num, uint8_t dec, int16_t x, int16_t y, uint8_t divider, int16_t length, uint8_t filler)
function	Print a specified length floating-point number at the specified coordinates.
parameters	num : the value to print(Supported range depends on the number of fractional digits used.Approx range is +/-2*(10^(9-dec))) dec : digits in the fractional part (1-5) 0 is not supported x : the x coordinate of the top left corner of the first number/sign y : the y coordinate of the top left corner of the first number/sign

	divider : Single character to use as decimal point. Default is '.' length : minimum number of digits/characters (including sign) to display filler : filler character to use to get the minimum length. The character will be inserted in front of the number, but after the sign. Default is ' ' (space).
returned value	None
notes	You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen

definiens	void Draw_Char(int16_t x, int16_t y, uint8_t c, uint16_t color,uint16_t bg, uint8_t size, boolean mode)
function	Draw a character at the specified coordinates.
parameters	x : the x coordinate of the top left corner of the character y : the y coordinate of the top left corner of the character c : the character to print color : the color value of the character to print bg : the background color value of the character to print size : the size of the character to print mode : 0-no overlap 1-overlap
returned value	None
notes	If the mode is overlap,the background color setting of the text is invalid.

definiens	size_t write(uint8_t c)
function	Write a character to print
parameters	c : the character to print
returned value	The statue of writing. 1-successful
notes	None

definiens	int16_t Get_Display_Width(void) const
function	Get the width of the screen
parameters	None
returned value	the width of the screen
notes	None

definiens	int16_t Get_Display_Height(void) const
function	Get the height of the screen
parameters	None
returned value	the height of the screen
notes	None