

**DLC Display Co., Limited**

德爾西顯示器有限公司



MODEL No: DLC0430DHCS-2

TEL: 86-755-86029824

FAX: 86-755-86029827

E-MAIL: [sales@dlcdisplay.com](mailto:sales@dlcdisplay.com)

WEB: [www.dlcdisplay.com](http://www.dlcdisplay.com)



### Record of Revision

Date	Revision No.	Summary
2024-09-12	1.0	Rev 1.0 was issued

### 1. Scope

This data sheet is to introduce the specification of DLC0430DHCS-2 module. This display area contains 480(RGB) x 800.

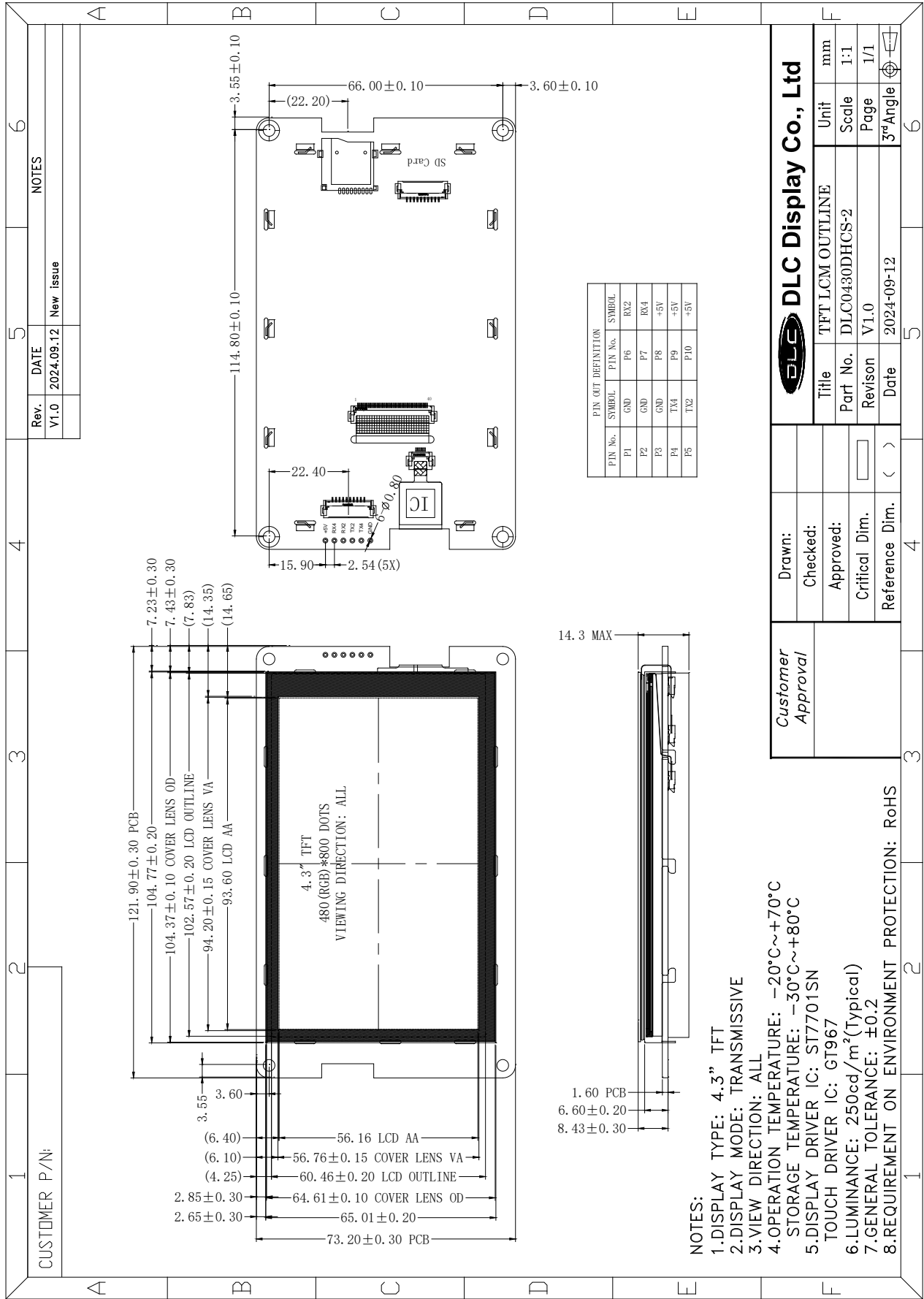
### 2. Application

Suitable for industry control, instrumentation, medical electronics, power electric equipment applications.

### 3. General Information

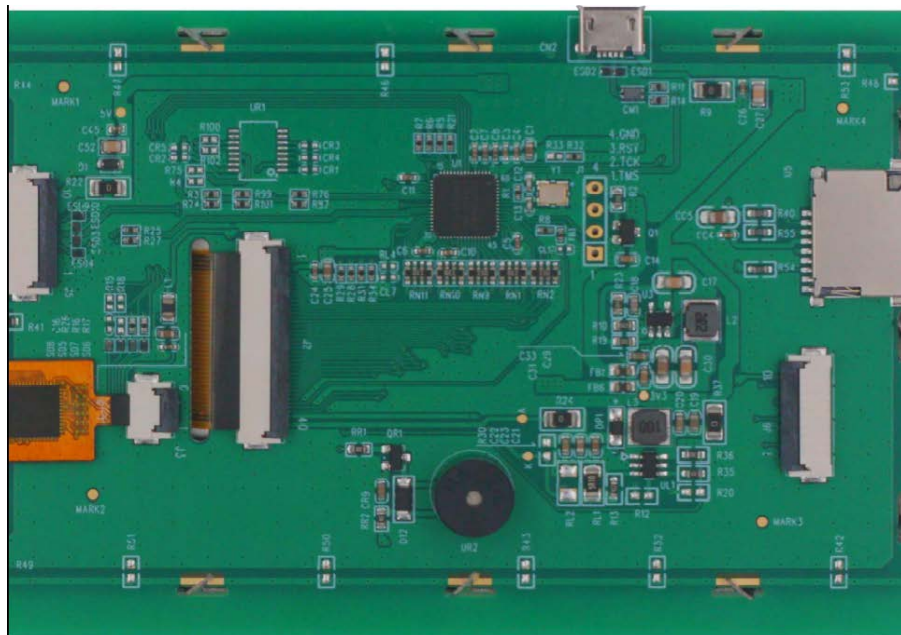
Item	Contents	Unit
Size	4.3	inch
Resolution	480(RGB) x 800	/
Interface	Serial UART	/
Technology type	IPS	/
Color Depth	16.7M color	/
Pixel pitch	0.117 x 0.117	mm
Pixel Configuration	R.G.B. Vertical Stripe	
Outline Dimension (W x H x D)	73.20 x 121.90 x 14.30Max.	mm
Active Area	56.16 x 93.60	mm
Display Driver IC	ST7701SN	/
Touch Driver IC	GT967	/
Viewing Direction	ALL	/
Backlight Type	LED	/
Weight	TBD	g

### 4. Outline Drawing



### 5. Interface signals

No	Symbol	Description	Remarks
1~3	GND	Ground	
4	TX4	Serial data (Customize)	
5	TX2	Serial data (Serial port to screen)	
6	RX2	Serial data (Serial port to screen)	
7	RX4	Serial data (Customize)	
8~10	+5V	Power supply voltage	

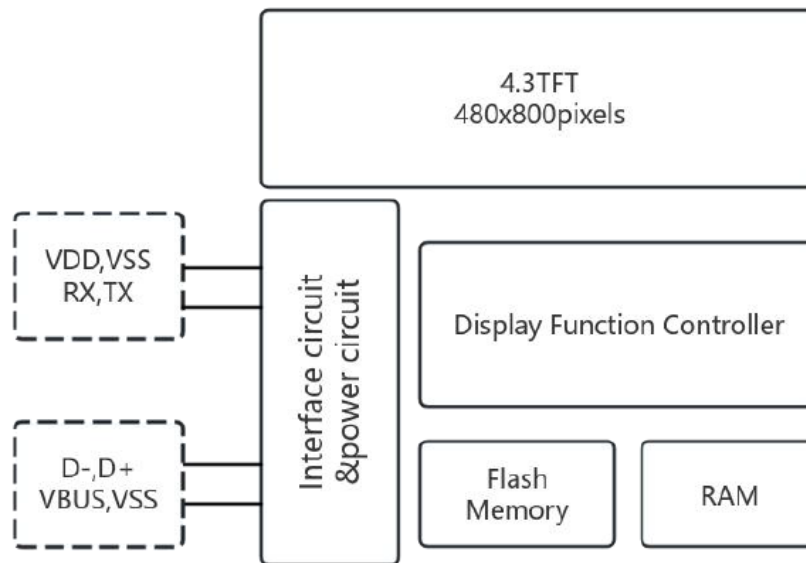


## 6. Electrical Specifications

### 6.1 Electrical Characteristics

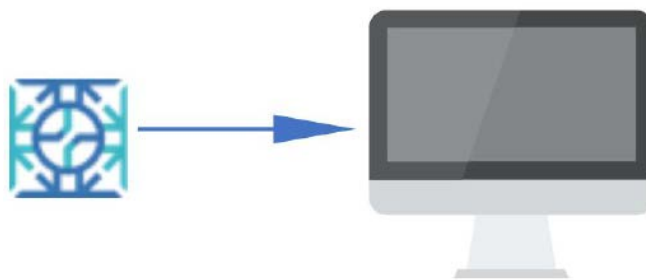
Parameter	Min.	Typ.	Max.	Unit
Power supply voltage	4.5	5.0	5.5	V
Operation temperature	-20	-	+70	°C
Storage temperature	-30	-	+80	°C

### 6.2 Block Diagram

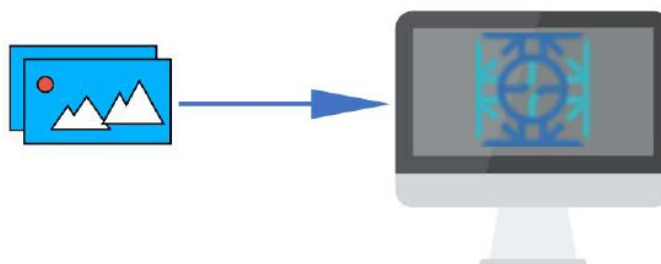


### 6.3 Quick Start Guide

1. Install AHMI-IDE on your computer



2. Import the image assets into AHMI-IDE and compile them to generate Texture.acf



3. Copy Texture.acf to USB or SD card and insert Mico USB or SD card to LCD Power on and upgrade UI



4. Once the upgrade is complete, unplug the Mico USB and power it on



5. Real-time communication via serial connection to the motherboard

## 7. AHMI Graphic display module serial communication protocol

### 7.1 Overview

#### 7.1.1 Scope of this document

The serial communication protocol is used for data interaction between the user baseboard and the graphic display module, allowing users to set and query LCD display content and module related parameters.

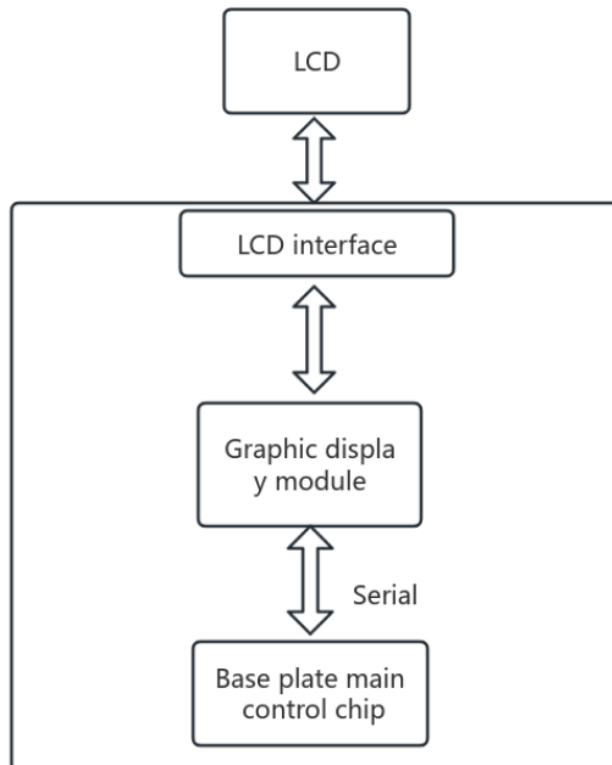
The purpose of this document is to describe how to build a user baseboard to control the graphic display and configure the graphic display module based on the serial communication protocol.

#### 7.1.2 Terms and abbreviations

Term or abbreviation	Characterization
Base plate	The user designs the hardware circuit that controls the graphic display module to drive the liquid crystal display content.
IDE	Self-developed simple and efficient tool for designing graphical display interfaces
UI	user interface
General purpose register	IDE Registers defined during software design engineering
Module register	Registers used in a specific range to set or read system parameters unrelated to the graphical interface, etc.

### 7.2 System structure

The backplane and the graphic display module communicate through the serial port. After the graphic display module receives the control message, it converts it into an LCD screen driving signal to drive the screen to display the specified content.



### 7.3 Serial port parameter settings

Classification	Parameter
Baud rate	Default 115200
Data bits	8
Check	None
Stop bit	1

### 7.4 Serial communication message frame description

#### 7.4.1 Message frame format

The general format of the message frame is as follows:

Frame header	Protocol version code	Request reply flag	Function code	Number of operating registers	Data	Check	End of frame
1 byte	0-1 byte	1byte			Maximum 484bytes	0-2 byte	1byte
		1bit(bit7)	3bits(bit6- 4)	4bits(bit3- 0)			
0xA5 /0x5A	0x00-0xFF	0-1	0-7	0-15	-	0x0000-0xFFFF	0xC3

**Frame header:** Used to identify the start of a frame of data, a fixed value of 0xA5 or 0x5A

0xA5: Indicates that the message transmission direction is sent from the base board to the graphic display module

0x5A: Indicates that the message transmission direction is from the graphic display module to the backplane.

**Protocol version code:** Whether to enable it can be configured in the IDE software. When enabled, it occupies 1 byte. It is not enabled by default. When enabled, it is used to assist the software to be compatible with older versions of the protocol, so that after the serial port protocol is upgraded, the backplane program does not need to adapt to the latest protocol due to compatibility issues. When not enabled, the core board program defaults to the baseboard using the latest serial port protocol. Please refer to the version history for the protocol version code.

**Request reply flag:** Identifies whether the backplane requests the graphics display module to reply when a message is received. When setting a request reply, if the message sent by the baseboard is a setting message (such as setting register, setting string register), the graphic display module returns a reply status information frame; if the message sent by the baseboard is a read message (such as reading a register) , the reply status information frame will be returned only when there is an error in the message received by the graphic display module (such as a verification error).

0: No reply requested

1: Request reply

**function code (bit 6 – 4):** Functions used to identify messages.

Bit 6 - 4	The baseboard sends a message to the graphics display module	Graphical display module replies to message base
-----------	--	--

0	Set value register	Reply numeric register value
1	Set string register	reply string register
2	Read string register	Reply status message
3	reserved	Reply long string register
4	Read numeric register	reserved
5	Set long string register	reserved
6	Read long string register	reserved
7	Reserved for expansion	reserved

**Number of operating registers (bit 3 – 0):** It is used to identify the number of registers to be operated on in the data field. The corresponding relationship is as follows::

Number of operating registers	Corresponding to the number of registers to be operated on
0	1
1	2
.....	.....
15	16

**Data:** The actual effective content of the message frame differs according to the data content format of different function codes. Please refer to the following introduction for details.

**Check:** You can configure whether to enable it in the IDE software. If verification is enabled, the verification data range is: the data after the frame header (not included) and before the verification (not included). Supports the following verification methods:

1. Not enabled
2. CRC check: 2 bytes, low byte first, high byte last. Using CRC-16/MODBUS (x16+x15+x2+1) parametric model. See Appendix 1 for CRC generation and verification related codes.
3. Checksum check: It occupies 2 bytes, with the low byte first and the high byte last.

The code related to Checksum generation and verification is shown in Appendix 2.

**End of frame:** Used to identify the end of message frame transmission, allowing the receiving end to quickly identify the end of frame transmission. Use the fixed value 0xC3 for identification. The detailed message frame format differs according to different functions, as follows:

#### 7.4.1.1 Set value register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>Request reply flag</b>	1byte	1bit(bit7) 0: No reply required 1: reply required
<b>Function code</b>		3bits(bit6-4) 0: Set numerical register
<b>Number of operating</b>		4bits(bit3-0)
<b>data</b>	<b>Register 1 ID</b>	2bytes
	<b>Register 1 value</b>	4bytes
	.....	.....
	<b>Register N ID</b>	2bytes

	<b>Register N value</b>	4bytes	
<b>Check</b>		0-2 byte	0x0000-0xFFFF
<b>End of frame</b>		1byte	0xC3

One frame of data can support setting up to 16 registers. The value in the number of operation registers should match the number of registers contained in the data. Message frame data format parsing example:

Example 1: A5 80 00 02 00 00 00 04 E2 CB C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
80	1	Request reply
	000	Set value register
	0000	Set 1 register
0002	The numerical register ID to be set is 2	
00000004	The value of the numerical register to be set is 4	
E2CB	The CRC value is 0xE2CB	
C3	end of frame	

Example 2: A5 02 03 E8 05 F5 E0 FF 00 64 00 00 00 64 03 E0 00 00 E0 FF 66 B4 C3 (hexadecimal)					
A5	Send data from backplane to graphics display module				
02	0	Request reply			
	000	Set value register			
	0010	Set 3 registers			
03 E8	The ID of the numerical register to be set is 1000	00 64	The ID of the numerical register to be set is 100	03 E0	The ID of the numerical register to be set is 992
05 F5 E0 FF	The value of the numerical register to be set is 99999999	00 00 00 64	The value of the numerical register to be set is 100	00 00 E0 FF	The value of the numerical register to be set is 57599
66 B4	CRC value is 0x66 B4				
C3	end of frame				

#### 7.4.1.2 Set string register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>request reply flag</b>	1byte	1bit(bit7) 0: No reply required 1: reply required
<b>function code</b>		3bits(bit6-4) 1: Set string register
<b>Number of operating registers</b>		4bits(bit3-0) 0

<b>data</b>	<b>total command length</b>	1byte	
	<b>Register ID</b>	2bytes	
	<b>Register sets the number of</b>	1byte	
	<b>Register setting</b>	1-32bytes	
<b>Check</b>		0-2 byte	0x0000-0xFFFF
<b>End of frame</b>		1byte	0xC3

The total command length is the data byte length occupied by all subsequent set string register commands.

The number of operating registers is forced to 0.

When setting the string register, you need to set the IDE project encoding format in advance: ASCII, UTF-8 or GB2312. The project default setting is ASCII encoding. Among them, ASCII format can only display English letters and cannot display Chinese characters. UTF-8 and GB2312 can display Chinese and English characters.

Message frame data format parsing example:

Example 1: (UTF-8 Encoding shows Chinese) : A5 90 0C 00 0F 09 E7 9D BF E6 99 BA E5 A8 81 22 F9 C3 (hexadecimal)		
A5		Send data from backplane to graphics display module
90	1	Request reply
	001	Set string register
	0000	Set 1 register
0C		The total length of the command is 12 bytes
000F		The string register ID to be set is 15
09		The number of bytes in the string to be set is 9
E7 9D BF E6 99 BA E5 A8 81		The string to be set is "DLC"(UTF-8 encoding format)
22 F9		The CRC value is 0x22 F9
C3		end of frame

Example 2: (GB2312 Encoding shows Chinese) : A5 90 09 00 0F 06 EE A3 D6 C7 CD FE 82 7B C3 (hexadecimal)		
A5		Send data from backplane to graphics display module
90	1	Request reply
	001	Set string register
	0000	Set 1 register
09		The total length of the command is 9 bytes
000F		The string register ID to be set is 15
06		The number of bytes in the string to be set is 6
EE A3 D6 C7 CD FE		The string to be set is "DLC"(GB2312 encoding format)
82 7B		The CRC value is 0x82 7B
C3		end of frame

Example 3: (UTF-8 Encoding display in Chinese and English) : A5 10 12 00 11 0F 68 65 6C 6C 6F EF BC 8C E7 9D BF E6 99 BA E5 A8 81 B0 24 C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
10	0	No reply requested
	001	Set string register
	0000	Set 1 register
12	The total length of the command is 18 bytes	
0011	The string register ID to be set is 17	
0F	The number of bytes in the string to be set is 15	
68 65 6C 6C 6F EF BC 8C E7 9D BF E6 99 BA E5 A8 81	The string to be set is "hello,DLC"(UTF-8 encoding format)	
B0 24	The CRC value is 0xB0 24	
C3	end of frame	

#### 7.4.1.3 Read string register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>reserved</b>	1byte	1bit(bit7) 0
<b>function code</b>		3bits(bit6-4) 2: Read string register
<b>Number of operating registers</b>		4bits(bit3-0) 0
<b>Data</b>		<b>Register ID</b>
	<b>Read string bytes</b>	1byte
<b>Check</b>	0-2 byte	0x0000-0xFFFF
<b>End of frame</b>	1byte	0xC3

The number of operating registers is forced to 0。

When the number of bytes in the read string is greater than 32 bytes, it will be processed as 32 bytes.

Data frame data format parsing example:

Example 1: A5 A0 00 0A 16 A5 4A C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
A0	1	Request reply
	010	Read string register
	0000	Read 1 register
000A	The string register ID to be read is 10	
16	The length of the string to be read is 22 bytes	
A5 4A	The CRC value is 0xA5 4A	
C3	end of frame	

#### 7.4.1.4 Read numeric register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>reserved</b>	1byte	1bit(bit7) 0

<b>function code</b>			3bits(bit6-4)	4: Read the numerical register
<b>Number of operating</b>			4bits(bit3-0)	0
<b>Data</b>	<b>Register 1 ID</b>	2bytes		
	<b>Register 2 ID</b>	2bytes		
	<b>-----</b>	.....		
	<b>Register N ID</b>	2bytes		
<b>Check</b>		0-2 byte	0x0000-0xFFFF	
<b>End of frame</b>		1byte	0xC3	

One frame of data can support reading up to 16 registers. The value in the number of operation registers should match the number of registers contained in the data.。

Data frame data format parsing example:

Example 1: A5 C0 00 02 F0 3D C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
C0	1	Request reply
	100	Read numeric register
	0000	Read 1 register
0002	The ID of the numerical register to be read is 2	
F0 3D	The CRC value is 0xF0 3D	
C3	end of frame	

Example 2: A5 42 00 02 00 64 03 E8 41 A5 C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
42	0	No reply request
	100	Read numeric register
	0010	Read 3 registers
00 02	The ID of the numerical register to be read is 2	
00 64	The ID of the numerical register to be read is 100	
03 E8	The ID of the numerical register to be read is 1000	
41 A5	The CRC value is 0x41 A5	
C3	end of frame	

#### 7.4.1.5 Set long string register

<b>Frame header</b>		1byte	0xA5/0x5A
<b>Protocol version code</b>		0-1 byte	
<b>reserved</b>	1byte	1bit(bit7)	0: No reply required 1: reply required
		<b>function code</b>	3bits(bit6-4)
<b>Number of operating</b>		4bits(bit3-0)	0
<b>Data</b>	<b>Register ID</b>	2bytes	
	<b>Register sets the number of bytes in the string</b>	1byte	
	<b>Register setting</b>	1-480bytes	



Check	0-2 byte	0x0000-0xFFFF
End of frame	1byte	0xC3

The number of operation registers is forced to 0, that is, one frame of data only supports setting 1 register. Setting the long string register is mainly used for strings or QR codes whose number of characters to be set is greater than 32 bytes, etc.

When the request reply flag is 1, that is, when a reply is required, the core board replies with a "reply to status information containing ID" frame

Message frame data format parsing example:

Example 1: (UTF-8 encoding displays Chinese) A5 D0 00 0F 00 2D E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 4B 67 C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
D0	1	Request reply
	101	Set long string register
	0000	Set 1 register
000F	The string register ID to be set is 15	
002D	The number of bytes in the string to be set is 45	
E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81 E7 9D BF E6 99 BA E5 A8 81	The string to be set is "DLC"  (UTF-8 encoding format)	
4B67	The CRC value is 0x4B67	
C3	end of frame	

Example 2: (QR code) : A5 50 00 11 00 5F 68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C F1 82 C3 (hexadecimal)		
A5	Send data from backplane to graphics display module	
5	0	No reply Request
	101	Set long string register
	0000	Set 1 register
0011	The string register ID to be set is 17	
005F	The number of bytes in the string to be set is 95	

68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C	The string to be set is " <a href="https://docs.graphichina.com/gao-jie-she-ji-liu-cheng/xu-ni-jian-pan-yu-gao-li-ang-xian-shi.html">https://docs.graphichina.com/gao-jie-she-ji-liu-cheng/xu-ni-jian-pan-yu-gao-li-ang-xian-shi.html</a> " (UTF-8 encoding format)
F1 82	The CRC value is 0x F1 82
C3	end of frame

7.4.1.6 Read long string register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>reserved</b>	1byte	1bit(bit7) 0
<b>function code</b>		3bits(bit6-4) 6: Read long string register
<b>Number of operating</b>		4bits(bit3-0) 0
<b>Data</b>	<b>Register ID</b>	2bytes
	<b>Read string bytes</b>	2bytes
<b>Check</b>	0-2 byte	0x0000-0xFFFF
<b>End of frame</b>	1byte	0xC3

The number of operating registers is forced to 0.

When the number of bytes in the read string is greater than 480 bytes, it will be processed as 480 bytes.

Data frame data format parsing example:

Example 1: A5 E0 00 0A 00 16 04 1A C3 (hexadecimal)	
A5	Send data from backplane to graphics display module
E0	1 Request reply
	110 Read long string register
	0000 Read 1 register
000A	The string register ID to be read is 10
0016	The length of the string to be read is 22 bytes
041A	The CRC value is 0x041A
C3	end of frame

7.4.1.7 Reply value register

<b>Frame header</b>	1byte	0xA5/0x5A
<b>Protocol version code</b>	0-1 byte	
<b>request reply flag</b>	1byte	1bit(bit7) 0



<b>function code</b>			3bits(bit6-4)	0: Reply register value
<b>Number of operating</b>			4bits(bit3-0)	
<b>data</b>	<b>Register 1 ID</b>	2bytes		
	<b>Register 1 value</b>	4bytes		
	.....	.....		
	<b>Register N ID</b>	2bytes		
	<b>Register N value</b>	4bytes		
<b>Check</b>		0-2 byte		0x0000-0xFFFF
<b>End of frame</b>		1byte		0xC3

One frame of data can support replying to a maximum of 16 registers. The value in the number of operation registers should match the number of registers contained in the data. In touch screen applications, when the touch serial port data is set to be sent out, the touch control will send the corresponding register and its value to the serial port output through the "reply register" message.

Message frame data format parsing example:

Example 1: 5A 00 00 02 00 00 00 04 63 03 C3 (hexadecimal)	
5A	The graphic display module returns data to the bottom board
00	0 reserved
	000 Reply register
	0000 Reply 1 register
0002	Reply register ID is 2
00000004	The value of the numerical register to be set is 4
63 03	The CRC value is 0x63 03
C3	end of frame

Example2 : 5A 02 03 E8 05 F5 E0 FF 00 64 00 00 00 64 03 E0 00 00 E0 FF 66 B4 C3 (hexadecimal)	
5A	The graphic display module returns data to the bottom board
02	0 reserved
	000 Reply register
	0010 Reply 3 registers
03E8	The reply register ID is 1000
05 F5 E0FF	The reply register value is 99999999
00 64	The reply register ID is 100
00000064	The reply register value is 100
03 E0	The reply register ID is 992
0000E0 FF	The reply register value is 57599
66B4	The CRC value is 0x66B4
C3	end of frame

7.4.1.8 Reply string register

<b>Frame header</b>	1byte	0x5A
<b>Protocol version code</b>	0-1 byte	

<b>request reply flag</b>		1bit(bit7)	0
<b>function code</b>		1byte	3bits(bit6-4) 1: Reply string register
<b>Number of operating registers</b>			
<b>data</b>	<b>Register ID</b>	2bytes	
	<b>Register sets the number of bytes in the string</b>	1byte	
	<b>Register setting string</b>	1-32bytes	
<b>Check</b>		0-2 byte	0x0000-0xFFFF
<b>End of frame</b>		1byte	0xC3

One frame of data can support replying to a maximum of 16 registers. The value in the number of operation registers should match the number of registers contained in the data. In touch screen applications, when the touch serial port data is set to be sent out, the touch control will send the corresponding register and its value to the serial port output through the "reply register" message. Message frame data format parsing example:

Example 1: 5A 10 00 0A 16 31 2E 36 2E 33 5F 62 75 69 6C 64 5F 30 33 2E 32 35 2E 30 39 2E 34 64 2C C3 (hexadecimal)		
5A		The graphic display module returns data to the bottom board
10	0	reserved
	001	reply string register
	0000	Reply 1 register
000A		The reply string register ID is 10
16		The number of bytes in the reply string is 22
31 2E 36 2E 33 5F 62 75 69 6C 64 5F 30 33 2E 32 35 2E 30 39 2E 34		Version number, corresponding string" 1.6.3_build_03.25.09.4"
642C		The CRC value is 0x642C
C3		End of frame

#### 7.4.1.9 Reply status message

<b>Frame header</b>		1byte	0x5A
<b>Protocol version code</b>		0-1 byte	
<b>request reply flag</b>		1byte	1bit(bit7) 0
<b>function code</b>			3bits(bit6-4) 2:Reply with detailed information
<b>Number of operating</b>			4bits(bit3-0)
<b>data</b>	<b>reserved</b>	1byte	
	<b>frame error code</b>	1byte	
	<b>Register 1 ID</b>	2bytes	
	<b>Register 1 operation error code</b>	1byte	
	.....	.....	



	<b>Register N ID</b>	2bytes	
	<b>Register N operation error code</b>	1byte	
<b>Check</b>		0-2 byte	0x0000-0xFFFF
<b>End of frame</b>		1byte	0xC3

One frame of data can support replying to a maximum of 16 registers. The value in the number of operation registers should match the number of registers contained in the data. frame error code:

Bit 7: CRC Verification status, 0: verification successful, 1: verification failed;

Bit 6 – Bit 0: Reserved, assigned 0 by default.

Register operation error code:

Bit 7: The register cannot set errors, 0: no error, 1: error;

Bit 6: Parameter out of range error, 0: no error, 1: error;

Bit 5: Register undefined error, 0: no error, 1: error

Bit 4: Operation register type error, 0: no error, 1: error

Bit 3 – Bit 0: Reserved, assigned 0 by default. □

Note: Bit 7 - Bit 4 The corresponding error code is only for setting module registers, and the return value is always 0 when setting registers on the IDE.

Message frame data format parsing example:

Example 1: (Receive without error) 5A 21 00 00 03 E8 00 03 E9 00 41 15 C3 (hexadecimal)		
5A		The graphic display module returns data to the bottom board
21	0	reserved
	010	Reply with detailed status information
	0001	Reply 2 registers
00		reserved
00		Frame error code is 0: no error
03 E8		Register 1 ID
00		Register 1 operation error code is 0, no error
03E9		Register 2 ID
00		Register 2 operation error code is 0, no error
41 15		The CRC value is 0x41 15
C3		end of frame

Example 2: (Receive data CRC error) 5A 21 00 80 03 E8 00 03 E9 00 C0 DD C3 (hexadecimal)		
5A		The graphic display module returns data to the bottom board
21	0	reserved
	010	Reply with detailed status information
	0001	Reply 2 registers
00		reserved
80		Frame error code bit7 is 1: CRC check error
03 E8		Register 1 ID
00		Register 1 operation error code is 0, no error
03E9		Register 2 ID
00		Register 2 operation error code is 0, no error
C0 DD		The CRC value is 0xC0 DD
C3		end of frame

Example 3: (Register 2 setting parameter out of range error) 5A 21 00 00 03 E8 00 27 51 40 72 EE C3 (hexadecimal)		
5A	The graphic display module returns data to the bottom board	
21	0	reserved
	010	Reply with detailed status information
	0001	Reply 2 registers
00	reserved	
00	Frame error code is 0: no error	
03 E8	Register 1 ID	
00	Register 1 operation error code is 0, no error	
27 51	Register 2 ID	
40	The operation error code bit6 of register 2 is 1, and the setting parameter is out of range error.	
72 EE	The CRC value is 0x72 EE	
C3	end of frame	

#### 7.4.1.10 Reply long string register

<b>Frame header</b>	1byte	0x5A
<b>Protocol version code</b>	0-1 byte	
<b>request reply flag</b>	1byte	1bit(bit7) 0
<b>function code</b>		3bits(bit6-4) 3: Reply string register
<b>Number of operating</b>		4bits(bit3-0) 0
<b>data</b>	<b>Register ID</b>	2bytes
	<b>Register sets the number of bytes in the string</b>	2bytes
	<b>Register setting string</b>	1-480bytes
<b>Check</b>	0-2 byte	0x0000-0xFFFF
<b>End of frame</b>	1byte	0xC3

One frame of data only supports replying to 1 register.

In touch screen applications, when touching the serial port data to send out a string, if the string exceeds 32 bytes, the "Reply Long String Register" frame will be replied, otherwise, the "Reply String Register" frame will be replied.

Message frame data format parsing example:

Example 1 :5A 30 00 0A 00 2D 68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C A2 72 C3 (十六进制)
---

5A	30			000A	002D	68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C	A2 72	C3
	0	011	0000					

Example 1 : 5A 30 00 0A 00 2D 68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C A2 72 C3 ( hexadecimal )

5A		The graphic display module returns data to the bottom board
3  0	0	reserved
	011	Reply long string register
	0000	Reply 1 string register
000A		The reply string register ID is 10
002D		The number of bytes in the reply string is 45
68 74 74 70 73 3A 2F 2F 64 6F 63 73 2E 67 72 61 70 68 69 63 68 69 6E 61 2E 63 6F 6D 2F 67 61 6F 2D 6A 69 65 2D 73 68 65 2D 6A 69 2D 6C 69 75 2D 63 68 65 6E 67 2F 78 75 2D 6E 69 2D 6A 69 61 6E 2D 70 61 6E 2D 79 75 2D 67 61 6F 2D 6C 69 61 6E 67 2D 78 69 61 6E 2D 73 68 69 2E 68 74 6D 6C		Corresponding string <a href="https://docs.graphichina.com/gao-jie-she-ji-liu-cheng/xu-ni-jian-pa-n-yu-gao-liang-xian-shi.html">https://docs.graphichina.com/gao-jie-she-ji-liu-cheng/xu-ni-jian-pa-n-yu-gao-liang-xian-shi.html</a> "
A2 72		The CRC value is 0xA2 72
C3		end of frame

#### 7.4.2 Data packet transmission sequence

The message frame data is packaged and transmitted by bytes. The frame header is sent first and the frame tail is sent last.

When the intra-frame functional field data occupies more than 1 byte::

- ◆ Check field: low byte first, high byte last
- ◆ Other functional fields: high byte first, low byte last

Example: Set the value of numerical register 1000 to 2000, use CRC mode for verification, and the data after packaging is:

A5 80 03 E8 00 00 07 D0 F9 40 C3

Among them, 03E8 (hexadecimal, corresponding to decimal is 1000) indicates that the register ID is 1000 000007D0(Hexadecimal, corresponding to decimal is 2000) means the setting value is 2000 F940 ((hex) means the CRC value is 0xF940

## 7.5 Graphical display interface predefined general register list

The graphical display interface register refers to the operating unit that is bound to each control and can be controlled by software when creating a graphical display interface through an IDE tool.

Predefined register allocation table:

serial numb	Register ID	content	Operation type	Type	Number of bytes	Default value	Remark
1	1	video	Read and write	numerical value	4	/	
2	2	Current page serial number	Read and write	numerical value	4	/	
3	3	Pass key encoding	Read and write	numerical value	4	/	
4	4	Clock variable year month day	Read and write	numerical value	4	/	
5	5	Clock variable hours minutes seconds	Read and write	numerical value	4	/	
6	6	buzzer	Read and write	numerical value	4	/	
7	7	backlight	Read and write	numerical value	4	/	
8	8	Frame rate	read only	numerical value	4	/	
9	9	touch calibration	Read and write	numerical value	4	/	
10	10	All-in-one software version	read only	numerical value	4	/	
11	11	IDE version	read only	String	22	/	
12	12	CPU0(USERCPU) version	read only	String	23	/	

13	13	CPU1(LCD CPU) version	read only	String	22	/	
14	14	devise serial number	read only	String	32	/	

## 7.6 Graphic display module register definition

The graphics display module register refers to the operating unit related to the graphics display module software and hardware, which can be set and queried by external devices (baseboard) to set and query software and hardware parameters and functions.。

The register IDs that qualify this section are defined in the range 10001 - 20000. The registers in this part will be checked for setting ranges. If they are out of range, no processing will be done. The parameters will only be identified when replying to the status information.

Out of range error. In addition, when executing a set command for a register that cannot be set, a register cannot be set error will also be identified when replying to the status information.。

The following is a uniformly defined graphic display module register definition.:

Serial number	Register ID	Register type	Operation type	content	Characterization	Default value	Remark
1	10000-10050	/	/	unavailable	/	/	For FPGA architecture software , GC9003 chip architecture software is not available
2	10051	string	read only	IDE version	/	/	
3	10052	string	read only	All-in-one software version			
4	10053	string	read only	Cpu1version			
5	10054	string	read only	Cpu0version			
6	10055	string	read only	devise serial number			
7	10056	numerical value	write only	system reset	0xAA5555AA Valid, other values are invalid		



8	10057	numerical value	Write only	Start touch calibration	0xAA5555AA Valid, other values are invalid		
9	10058	numerical value	Read and write	RTC date			Valid only when RTC is enabled
10	10059	numerical value	Read and write	RTC time			Valid only when RTC is enabled
11	10060	numerical value	Read and write	RTC Week	Day of the week, range 1-7		Valid only when RTC is enabled
12	10061	numerical value	Read and write	Touch audio	0: off; 1: on		Valid only when touch audio function is enabled
13	10062	numerical value	write only	buzzer alarm	0: Stop alarm, other values: alarm time, unit 10ms		
14	10063	numerical value	write only	Music playback control	High 8 bits: control mode, 1: play; 2: pause; 3: resume; 4: stop Lower 24 bits: audio ID, range: 0-255		For example: 0x010000 02: Play audio with serial number 2
15	10064	numerical value	Read and write	RTC-Year	Year, range: 2000-2099		Valid only when RTC is enabled
16	10065	numerical value	Read and write	RTC-Month	Month, range: 1-12		Valid only when RTC is enabled
17	10066	numerical value	Read and write	RTC-Day	Day, range : 1-31		Valid only when RTC is enabled
18	10067	numerical value	Read and write	RTC-Hour	Hour, range : 0-23		Valid only when RTC is enabled
19	10068	numerical value	Read and write	RTC-Minute	Minute, range : 0-59		Valid only when RTC is enabled



20	10069	numerical value	Read and write	RTC-Second	Second, range : 0-59		Valid only when RTC is enabled
21	10070	/	/	reserved			

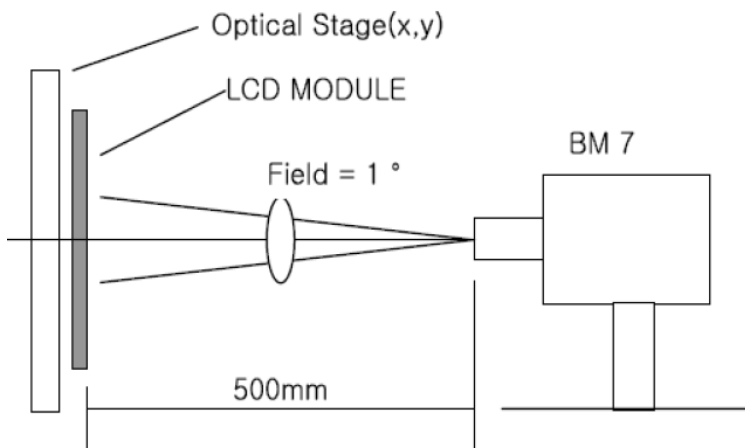
## 8. Optical Specification

Item	Symbol	Condition	Min	Typ.	Max.	Unit	Remark		
Contrast Ratio	CR	$\theta=0^\circ$	-	500	--		Note1 Note2		
Response Time	TR+TF	25°C	--	30	40	ms	Note1 Note3		
View Angles	$\Theta T$	$CR \geq 10$	-	80	--	Degree	Note 4		
	$\Theta B$		-	80	--				
	$\Theta L$		-	80	--				
	$\Theta R$		-	80	--				
Chromaticity	White	Brightness is on	Typ-0.05	Typ+0.05	TBD	TBD	Note5, Note1		
								Red	x
	y								TBD
	Green							x	TBD
								y	TBD
	Blue							x	TBD
								y	TBD
	Luminance							L	
Uniformity (White)	U		80	--	--	%	Note1 Note7		

Note 1: Definition of optical measurement system.

Temperature = 25°C(±3°C)

LED back-light: ON, Environment brightness < 150 lx

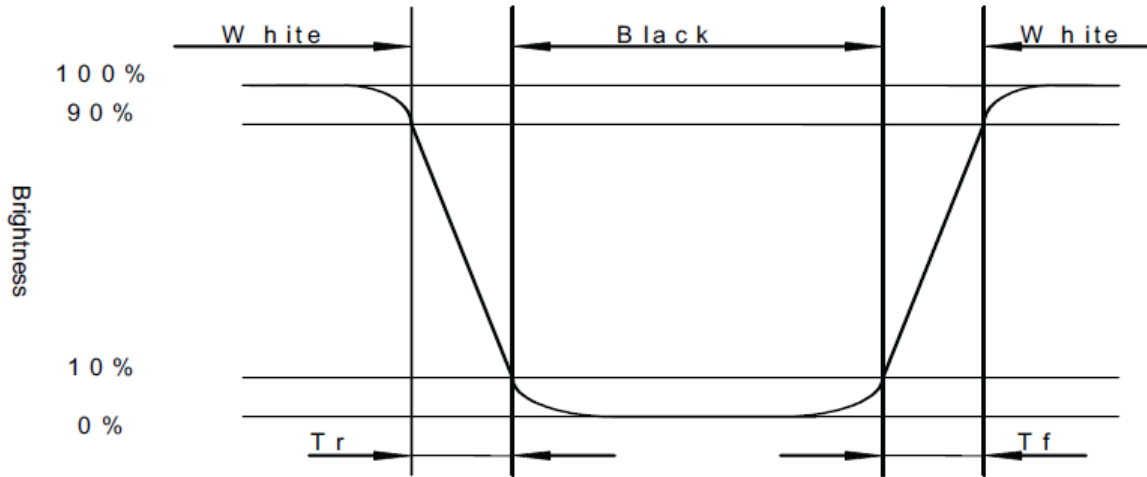


Note 2: Contrast ratio is defined as follow:

$$\text{Contrast Ratio} = \frac{\text{Surface Luminance with all white pixels}}{\text{Surface Luminance with all black pixels}}$$

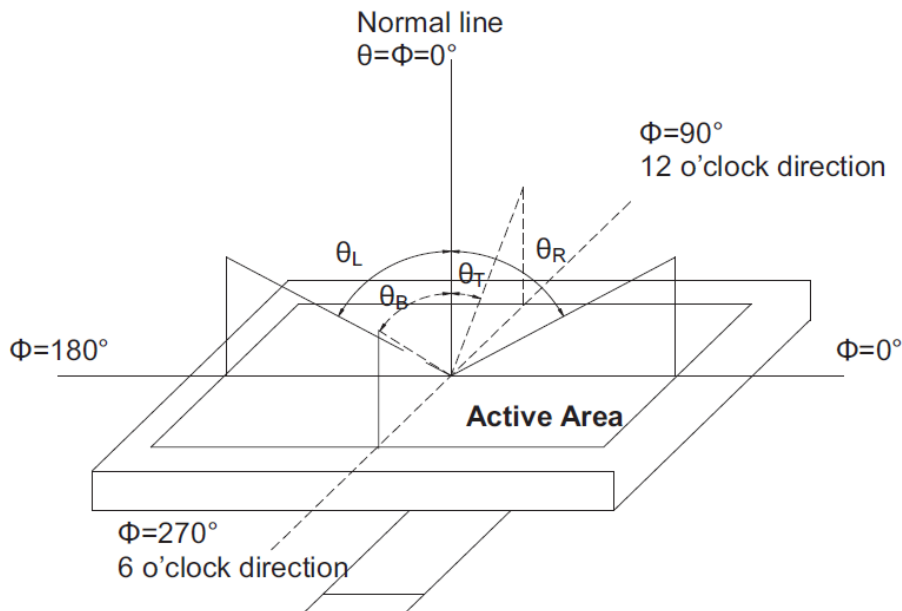
Note 3: Response time is defined as follow:

Response time is the time required for the display to transition from black to white (Rise Time,  $T_r$ ) and from white to black (Decay Time,  $T_f$ ).



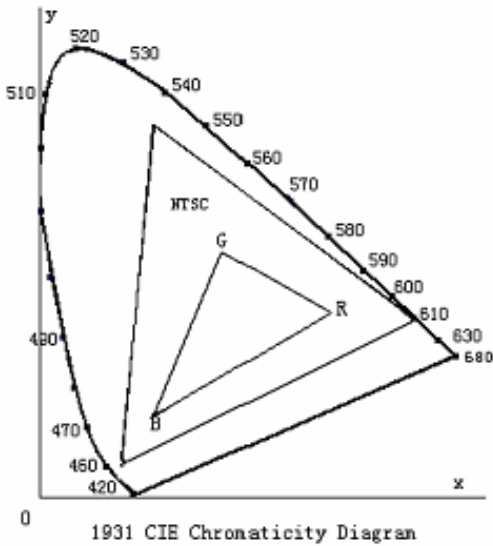
Note 4: Viewing angle range is defined as follow:

Viewing angle is measured at the center point of the LCD.



Note 5: Color chromaticity is defined as follow: (CIE1931)

Color coordinates measured at center point of LCD.



$$S = \frac{\text{area of RGB triangle}}{\text{area of NTSC triangle}} \times 100\%$$

Note 6: Luminance is defined as follow:

Luminance is defined as the brightness of all pixels “White” at the center of display area on optimum contrast.

Note 7: Luminance Uniformity is defined as follow:

Active area is divided into 9 measuring areas (Refer Fig. 2). Every measuring point is placed at the center of each measuring area.

$$\text{Uniformity}(U) = \frac{\text{Minimum Luminance(brightness) in 9 points}}{\text{Maximum Luminance(brightness) in 9 points}}$$

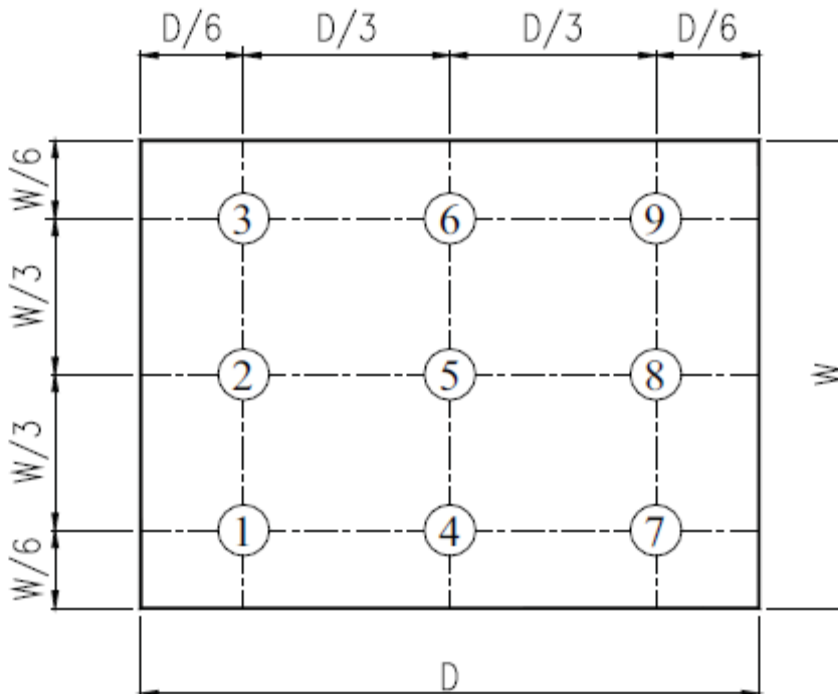


Fig. 2 Definition of uniformity

## 9. Environmental / Reliability Tests

No	Test Item	Condition	Judgment criteria
1	High Temp Operation	Ta= +70°C, 120hrs	Per table in below
2	Low Temp Operation	Ta= -20°C, 120hrs	Per table in below
3	High Temp Storage	Ts= +80°C, 120hrs	Per table in below
4	Low Temp Storage	Ts= -30°C, 120hrs	Per table in below
5	High Temp & High Humidity Storage	Ts= +40°C, 90% RH, 120 hours	Per table in below (polarizer discoloration is excluded)
6	Thermal Shock (Non-operation)	-30°C 30 min~+80°C 30 min, Change time:5min, 10 Cycles	Per table in below
7	ESD (Operation)	C=150pF, R=330Ω, 5points/panel Air:±8KV, 5times; Contact:±4KV, 5 times;	Per table in below
8	Vibration (Non-operation)	Frequency range:10~55Hz, Stroke:1.5mm Sweep:10Hz~55Hz~10Hz 2 hours for each direction of X.Y.Z.	Per table in below
9	Shock (Non-operation)	60G 6ms, ±X,±Y,±Z 3times, for each direction	Per table in below
10	Package Drop Test	Height:80 cm, 1 corner, 3 edges, 6 surfaces	Per table in below

INSPECTION	CRITERION(after test)
Appearance	No Crack on the FPC, on the LCD Panel
Alignment of LCD Panel	No Bubbles in the LCD Panel No other Defects of Alignment in Active area
Electrical current	Within device specifications
Function / Display	No Broken Circuit, No Short Circuit or No Black line No Other Defects of Display

## 10. Precautions for Use of LCD Modules

### 11.1 Safety

The liquid crystal in the LCD is poisonous. Do not put it in your mouth. If the liquid crystal touches your skin or clothes, wash it off immediately using soap and water.

### 11.2 Handling

- A. The LCD and touch panel is made of plate glass. Do not subject the panel to mechanical shock or to excessive force on its surface.
- B. Do not handle the product by holding the flexible pattern portion in order to assure the reliability
- C. Transparency is an important factor for the touch panel. Please wear clear finger sacks, gloves and mask to protect the touch panel from finger print or stain and also hold the portion outside the view area when handling the touch panel.
- D. Provide a space so that the panel does not come into contact with other components.
- E. To protect the product from external force, put a covering lens (acrylic board or similar board) and keep an appropriate gap between them.
- F. Transparent electrodes may be disconnected if the panel is used under environmental conditions where dew condensation occurs.
- G. Property of semiconductor devices may be affected when they are exposed to light, possibly resulting in IC malfunctions.
- H. To prevent such IC malfunctions, your design and mounting layout shall be done in the way that the IC is not exposed to light in actual use.

### 11.3 Static Electricity

- A. Ground soldering iron tips, tools and testers when they are in operation.
- B. Ground your body when handling the products.
- C. Power on the LCD module before applying the voltage to the input terminals.
- D. Do not apply voltage which exceeds the absolute maximum rating.
- E. Store the products in an anti-electrostatic bag or container.

### 11.4 Storage

- A. Store the products in a dark place at  $+25^{\circ}\text{C}\pm 10^{\circ}\text{C}$  with low humidity (40% RH to 60% RH). Don't expose to sunlight or fluorescent light.
- B. Storage in a clean environment, free from dust, active gas, and solvent.

### 11.5 Cleaning

- A. Do not wipe the touch panel with dry cloth, as it may cause scratch.
- B. Wipe off the stain on the product by using soft cloth moistened with ethanol. Do not allow ethanol to get in between the upper film and the bottom glass. It may cause peeling issue or defective operation. Do not use any organic solvent or detergent other than ethanol.

### 11.6 Cautions for installing and assembling

- A. Bezel edge must be positioned in the area between the Active area and View area. The bezel may press the touch screen and cause activation if the edge touches the active area. A gap of approximately 0.5mm is needed between the bezel and the top electrode. It may cause unexpected activation if the gap is too narrow. There is a tolerance of 0.2 to 0.3mm for the outside dimensions of the touch panel and tail. A gap must be made to absorb the tolerance in the case and connector.
- B. In order to make the display assembly stable and firm, DLC recommends to design some supporting at the display backside, especially for the display with tape-attached touch panel, such supporting is important and essential, or else, the display may drop-off from front after some period of time.
- C. Do not display the fixed pattern for a long time because it may develop image sticking due to the LCD structure. If the screen is displayed with fixed pattern, use a screen saver.

