

**SPECIFICATIONS FOR  
LIQUID CRYSTAL DISPLAY MODULE**

**MODEL NO : ACM053BWWB-H-U**

**CUSTOMER :**

**APPROVED SIGNATURE**

**DSGD :**

**CHKD : Peter**

**APPD : Peng Jun**

**DATE : Aug.21.2017**

**YUDU AMSON ELECTRONICS CO.,LTD.**

**YuDu Industrial Garden,Ganzhou City,JiangXi,China**

**TEL : 86-797-6330063**

**FAX : 86-797-6330055**



Model No: ACM053B

<u>Revision Record</u>				
No.	Date	Model No.	Version	Remarks
1	Aug.21.2017	ACM053BWVB-H-U	REV.0	Spec RoHs-Compliant



Model No: ACM053B

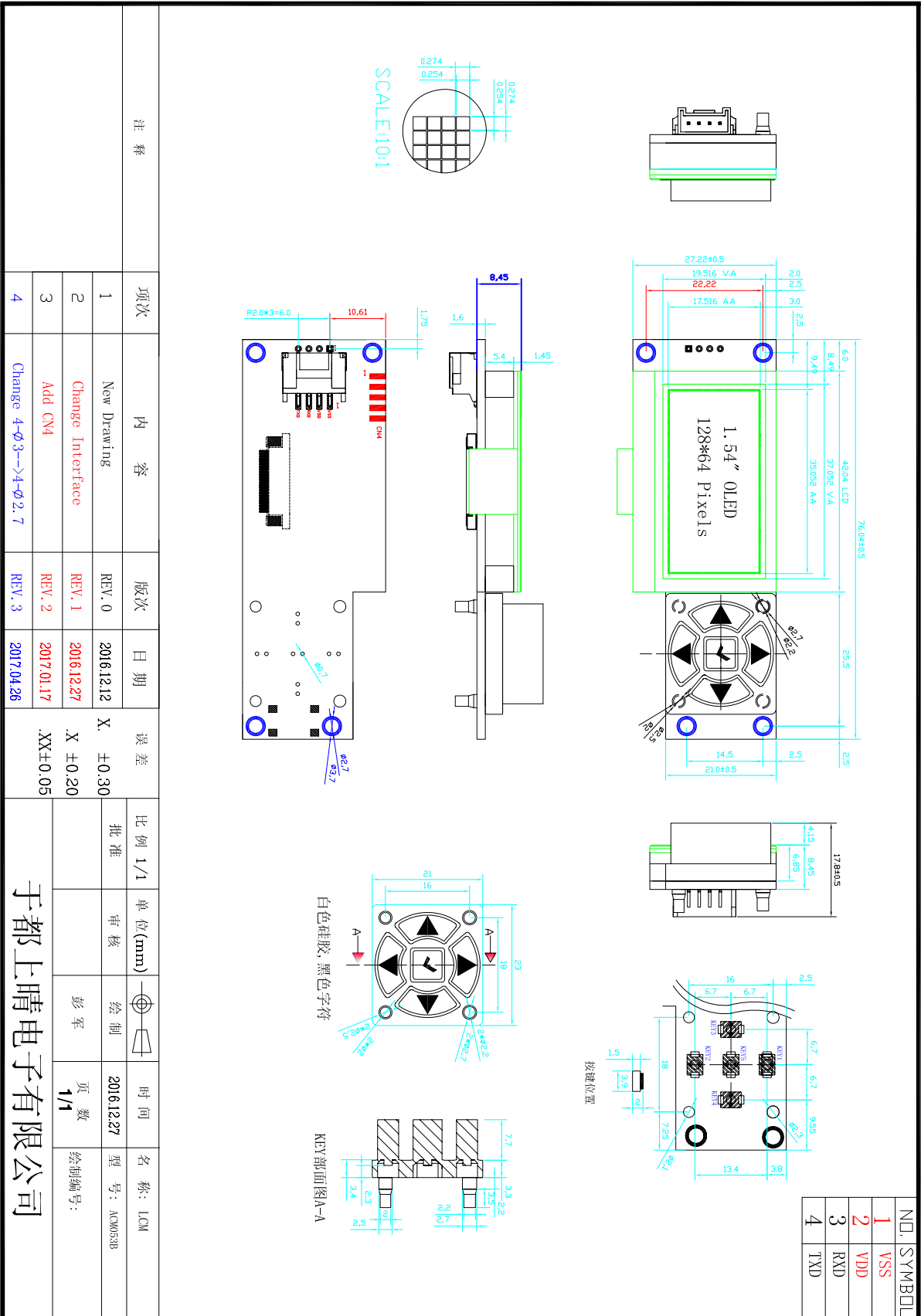
**GENERAL SPECIFICATION**

ITEM	DESCRIPTION				
Product No	ACM053BWWB-H-U				
OLED Type	■ OLED White & Black				
Rear Polarizer	■ Reflective / Positive				
Backlight Type	■ OLED				
OLED Color	<input type="checkbox"/> Yellow	<input type="checkbox"/> Green	<input type="checkbox"/> Amber	■ White	<input type="checkbox"/>
Temperature Range	■ Wide Temp., 5V, Single Supply Voltage				
Frame	<input type="checkbox"/> Black				

**TO BE VERY CAREFUL !**

The OLED driver ICs are made by CMOS process, which are very easy to be damaged by static charge, make sure the user is grounded when handling the LCM.

**✘ This parts comply with RoHs**





**ABSOLUTE MAXIMUM RATING**

(1) Electrical Absolute Ratings

Item	Symbol	Min.	Max.	Unit	Note
Power Supply for Logic	$V_{DD}-V_{SS}$	-0.3	5.5	Volt	
Input Voltage	$V_I$	-0.3	$V_{DD}$	Volt	

(2) Environmental Absolute Maximum Ratings

Item	Wide Temperature			
	Operating		Storage	
	Min.	Max.	Min.	Max.
Ambient Temperature	-40°C	+70°C	-40°C	+85°C
Humidity(without condensation)	Note 4,5		Note 4,6	

Note 4 Background color changes slightly depending on ambient temperature. This phenomenon is reversible.

Note 5  $T_a \leq 70^\circ\text{C}$ : 75RH max

$T_a > 70^\circ\text{C}$ : absolute humidity must be lower than the humidity of 75%RH at  $70^\circ\text{C}$

Note 6  $T_a$  at  $-30^\circ\text{C}$  will be <48hrs, at  $80^\circ\text{C}$  will be <120hrs when humidity is higher than 70%.



Model No: ACM053B

### ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ	Max.	Unit	note
Power Supply for Logic	$V_{DD}-V_{SS}$	-	4.5	5.0	5.5	Volt	
Input Voltage	$V_{IL}$	L level	0	-	$0.3 V_{DD}$	Volt	
	$V_{IH}$	H level	$0.7 V_{DD}$	-	$V_{DD}$	Volt	
Output Voltage	$V_{OL}$	L level	0	-	$0.3 V_{DD}$		
	$V_{OH}$	H level	$0.7 V_{DD}$	-	$V_{DD}$		
Power Supply Current for OLED	$I_{DD}$	$V_{DD}=3.3V$ $V_O-V_{SS}=12.5V$	-	35.8	-	mA	

### OPTICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ	Max.	Unit	note
Viewing angle range	$\Phi_f$ (12 o'clock)	$V_O-V_{SS}=12.5V$ $T_a=25^\circ C$	-	85	-	Degree	9,10
	$\Phi_b$ (6 o'clock)		-	85	-		
	$\Phi_l$ (9 o'clock)		-	85	-		
	$\Phi_r$ (3 o'clock)		-	85	-		
Rise Time	$T_r$	$V_O-V_{SS}=12.5V$ $T_a=25^\circ C$	-	300		nS	
Fall Time	$T_f$		-	300			
Frame frequency	F <sub>rm</sub>		-	-	-	Hz	8,10
Contrast	Cr		10000	-	-		7
Brightness	L		60	80	-	cd/m <sup>2</sup>	
Peak Emission Wavelength	C.I.E (White)		X=0.25 Y=0.27	X=0.29 Y=0.31	X=0.33 Y=0.35	nm	



Model No: ACM053B

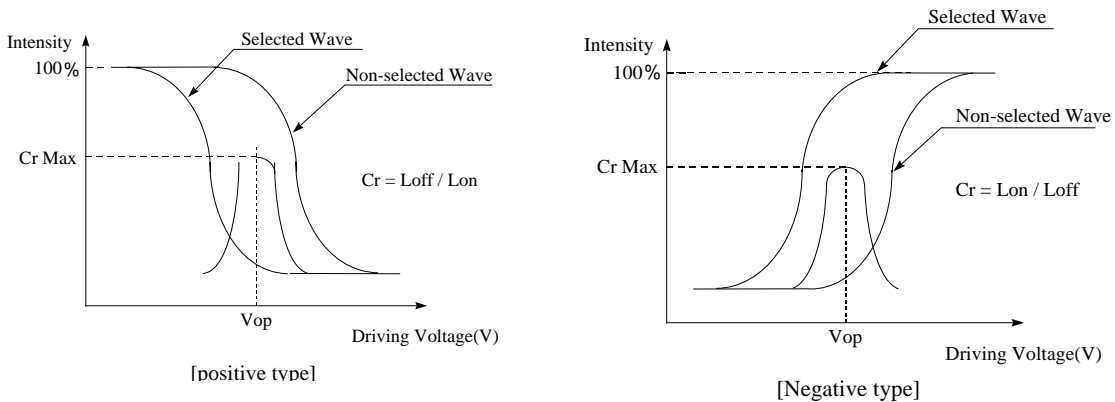
**MECHANICAL SPECIFICATION**

ITEM	DESCRIPTION
Product No.	ACM053BWWB-H-U
Viewing Area	37.052(W)mm×19.516(H)mm
Module Size	76.04(W)×27.22(H)×17.8 (D)
Dot Size	0.254(W)mm×0.254(H)mm
Dot Pitch	0.274(W)mm×0.274(H)mm
Display Format	128 (W)×64 (H)
Controller Interface	STM32F030C8T6 UART
ROM Selection	-
Built-in	With DC/DC Converter

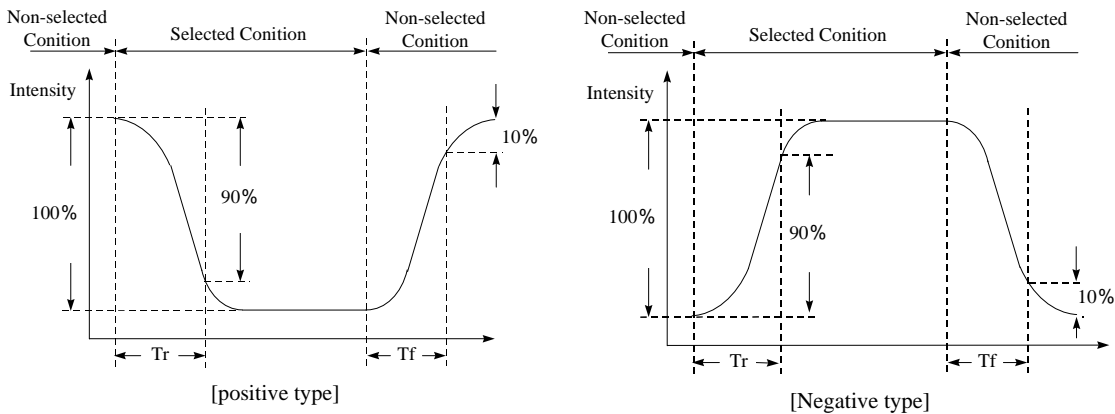
**INTERFACE PIN ASSIGNMENT(CN2)**

Pin No.	Pin Out	Level	Description
1	VSS	0V	Power Supply Ground
2	+5V	+5V	Power Supply Voltage
3	RXD	H/L	Serial Receive Signal
4	TXD	H/L	Serial Transmit Signal

**[Note 7] Definition of Operation Voltage (Vop)**



**[Note 8] Definition of Response Time (Tr, Tf)**



**Conditions:**

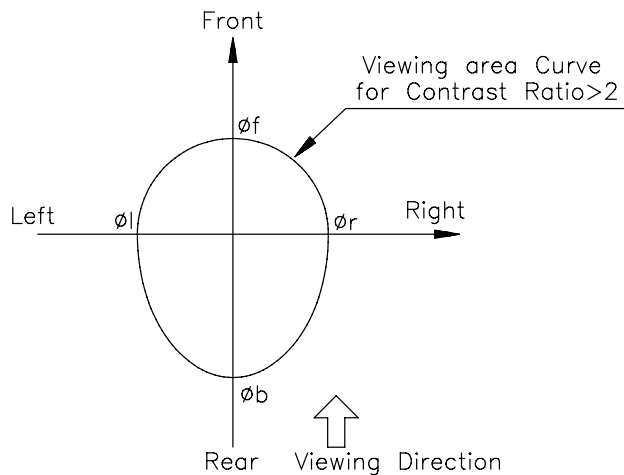
**Operating Voltage : Vop**

**Frame Frequency : 64 Hz**

**Viewing Angle( $\theta$  ,  $\phi$ ) :  $0^\circ$  ,  $0^\circ$**

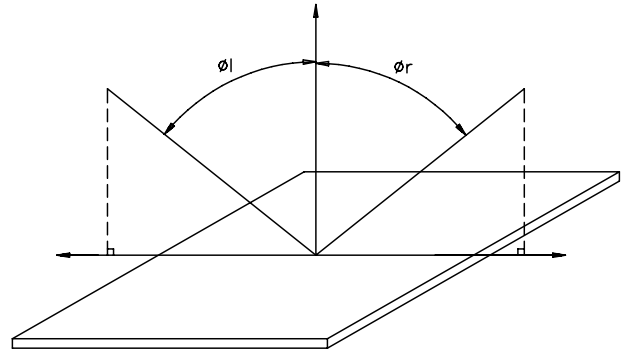
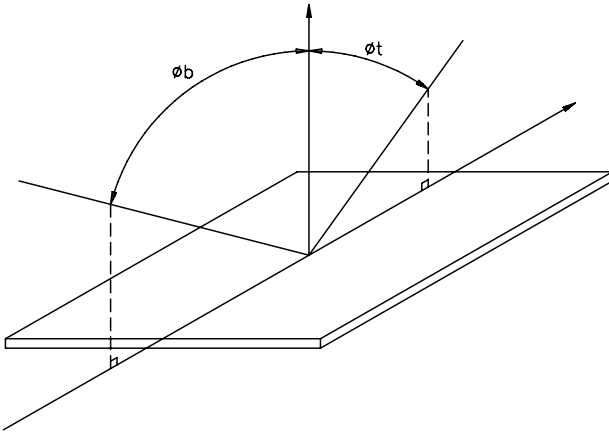
**Driving Wave form : 1/N duty, 1/a bias**

**[Note 9] Definition of Viewing Direction**





[Note 10] Definition of viewing angle





## FUNCTIONAL SPECIFICATION

### UART CONFIGURATION:

ITEM	SETTING VALUE
BAUD RATE	115200
DATA BIT	8
STOP BIT	1
PARITY BIT	NONE
FLOW CONTROL	NONE

### COMMAND LIST:

Code	Function	Sequence of HEX command mode through UART	API for Arduino
N/A	Sent a image(192X64 bitmap) to LED Display (An array consist of 1536 bytes bitmap information)	<ol style="list-style-type: none"> <li>1. A "for" loop to send 1536 bytes user define display information</li> <li>2. Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>for (i = 0 ; i &lt; 1536; i++) {   Serial.write(User_define_array[i]); } while (Serial.read() !='E') {} delay(2);</pre>
0x80	Write a 5X7 Character	<ol style="list-style-type: none"> <li>1. Send 0x80</li> <li>2. Send which line to put this character</li> <li>3. Send which cloumn to put this character</li> <li>4. Send character's ASCII code</li> <li>5. Wait until receive a module available byte ('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_5X7_Character( int line, int column, int negative, char Char) {   Serial.write(0x80);   Serial.write(line);   Serial.write(column);   Serial.print(Char);   while (Serial.read() !='E') {}   delay(2); }</pre>
0x81	Write a 8X8 String	<ol style="list-style-type: none"> <li>1. Send 0x81</li> <li>2. Send which line to start the string</li> <li>3. Send which cloumn to start the string</li> <li>4. Send string</li> <li>5. Wait until receive a module available byte('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_5X7_String( int line, int column, int negative, char * string) {   Serial.write(0x81);   Serial.write(line);   Serial.write(column);   Serial.print(string);   while (Serial.read() !='E') {}   delay(2); }</pre>
0x82	Write a 8X16 Character	<ol style="list-style-type: none"> <li>1. Send 0x82</li> <li>2. Send which line to put this character</li> <li>3. Send which cloumn to put this character</li> <li>4. Send character's ASCII code</li> <li>5.Wait until receive a module available byte('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_8X16_Character( int line, int column, int negative, char Char) {   Serial.write(0x82);   Serial.write(line);   Serial.write(column);   Serial.print(Char);   while (Serial.read() !='E') {}   delay(2); }</pre>



## Model No: ACM053B

0x83	Write a 8X16 String	<ol style="list-style-type: none"> <li>1. Send 0x83</li> <li>2. Send which line to start the string</li> <li>3. Send which column to start the string</li> <li>4. Send string</li> <li>5. Wait until receive a module available byte('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_8X16_String( int line, int column, int negative, char * string) {     Serial.write(0x83);     Serial.write(line);     Serial.write(column);     Serial.print(string);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x84	Display a 8X8 pattern	<ol style="list-style-type: none"> <li>1. Send 0x84</li> <li>2. Send the Up Left X coordinate of pattern</li> <li>3. Send the Up Left Y coordinate of pattern</li> <li>4. Send the ID of pattern</li> <li>5. Wait until receive a module available byte ('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_8X8_Pattern( int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {     Serial.write(0x84);     Serial.write(Up_Left_Xpos);     Serial.write(Up_Left_Ypos);     Serial.write(Pattern_ID);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x85	Display a 8X16 pattern	<ol style="list-style-type: none"> <li>1. Send 0x85</li> <li>2. Send the Up Left X coordinate of pattern</li> <li>3. Send the Up Left Y coordinate of pattern</li> <li>4. Send the ID of pattern</li> <li>5. Wait until receive a module available byte ('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_8X16_Pattern( int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {     Serial.write(0x85);     Serial.write(Up_Left_Xpos);     Serial.write(Up_Left_Ypos);     Serial.write(Pattern_ID);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x86	Display a 16X16 pattern	<ol style="list-style-type: none"> <li>1. Send 0x86</li> <li>2. Send the Up Left X coordinate of pattern</li> <li>3. Send the Up Left Y coordinate of pattern</li> <li>4. Send the ID of pattern</li> <li>5. Wait until receive a module available byte ('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_16X16_Pattern( int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {     Serial.write(0x86);     Serial.write(Up_Left_Xpos);     Serial.write(Up_Left_Ypos);     Serial.write(Pattern_ID);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x87	Display a 32X32 pattern	<ol style="list-style-type: none"> <li>1. Send 0x87</li> <li>2. Send the Up Left X coordinate of pattern</li> <li>3. Send the Up Left Y coordinate of pattern</li> <li>4. Send the ID of pattern</li> <li>5. Wait until receive a module available byte ('E') from LED Display</li> <li>6. Wait 2ms</li> </ol>	<pre>void Write_32X32_Pattern( int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {     Serial.write(0x87);     Serial.write(Up_Left_Xpos);     Serial.write(Up_Left_Ypos);     Serial.write(Pattern_ID);     while (Serial.read() !='E') {}     delay(2); }</pre>



## Model No: ACM053B

0x90	Draw a line	<ol style="list-style-type: none"> <li>1. Send 0x90</li> <li>2. Send the X coordinate of first point</li> <li>3. Send the Y coordinate of first point</li> <li>4. Send the X coordinate of second point</li> <li>5. Send the Y coordinate of second point</li> <li>6. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Draw_Line( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative ) {     Serial.write(0x90);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(X1_Pos);     Serial.write(Y1_Pos);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x91	Draw a Rectangle	<ol style="list-style-type: none"> <li>1. Send 0x91</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Draw_Rectangle( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative ) {     Serial.write(0x91);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(X1_Pos);     Serial.write(Y1_Pos);     Serial.write(0 or 1);     while (Serial.read() !='E') {} }</pre>
0x92	Draw a filled Rectangle	<ol style="list-style-type: none"> <li>1. Send 0x92</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Draw_Filled_Rectangle( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative ) {     Serial.write(0x92);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(X1_Pos);     Serial.write(Y1_Pos);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x93	Draw a Square	<ol style="list-style-type: none"> <li>1. Send 0x93</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the width of this square</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Square( int X0_Pos, int Y0_Pos, int width, int negative ) {     Serial.write(0x93);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(width);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>



## Model No: ACM053B

0x94	Draw a Circle	<ol style="list-style-type: none"> <li>1. Send 0x94</li> <li>2. Send the X coordinate of the center</li> <li>3. Send the Y coordinate of the center</li> <li>4. Send the radius of this circle</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Circle( int X0_Pos, int Y0_Pos, int radius, int negative ) {     Serial.write(0x94);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(radius);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x95	Draw a filled Circle	<ol style="list-style-type: none"> <li>1. Send 0x95</li> <li>2. Send the X coordinate of the center</li> <li>3. Send the Y coordinate of the center</li> <li>4. Send the radius of this circle</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Filled_Circle( int X0_Pos, int Y0_Pos, int radius, int negative ) {     Serial.write(0x95);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(radius);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x96	Draw a tip upward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x96</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the height of the tip to the bottom</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Triangle_Up_Ward( int X0_Pos, int Y0_Pos, int height, int negative ) {     Serial.write(0x96);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(height);     Serial.write(0 or 1);     while (Serial.read() !='E') {} }</pre>
0x97	Draw a filled tip upward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x97</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the height of the tip to the bottom</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Filled_Triangle_Up_Ward( int X0_Pos, int Y0_Pos, int height, int negative ) {     Serial.write(0x97);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(height);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x98	Draw a tip downward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x98</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the height of the tip to the top</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Triangle_Down_Ward( int X0_Pos, int Y0_Pos, int height, int negative ) {     Serial.write(0x98);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(height);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>



## Model No: ACM053B

0x99	Draw a filled tip downward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x99</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the height of the tip to the top</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw__Filled_Triangle_Down_Ward( int X0_Pos, int Y0_Pos, int height, int negative ) {     Serial.write(0x99);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(height);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x9a	Draw a tip leftward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x9a</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the width of the tip to the right</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Triangle_Left_Ward( int X0_Pos, int Y0_Pos, int width, int negative ) {     Serial.write(0x9a);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(width);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x9b	Draw a filled tip leftward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x9b</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the width of the tip to the right</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Filled_Triangle_Left_Ward( int X0_Pos, int Y0_Pos, int width, int negative ) {     Serial.write(0x9b);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(width);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x9c	Draw a tip rightward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x9c</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the width of the tip to the left</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Triangle_Right_Ward( int X0_Pos, int Y0_Pos, int width, int negative ) {     Serial.write(0x9c);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(width);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>
0x9d	Draw a filled tip rightward Triangle	<ol style="list-style-type: none"> <li>1. Send 0x9d</li> <li>2. Send the X coordinate of the tip</li> <li>3. Send the Y coordinate of the tip</li> <li>4. Send the width of the tip to the left</li> <li>5. Send 1 or 0 for display mode (1 for positive, 0 for negative)</li> <li>6. Wait until receive a module available byte ('E') from LED Display</li> <li>7. Wait 2ms</li> </ol>	<pre>void Draw_Filled_Triangle_Right_Ward( int X0_Pos, int Y0_Pos, int width, int negative ) {     Serial.write(0x9d);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(width);     Serial.write(0 or 1);     while (Serial.read() !='E') {}     delay(2); }</pre>



## Model No: ACM053B

0x9e	Set a pixel for positive display (show pixel)	<ol style="list-style-type: none"> <li>1. Send 0x9e</li> <li>2. Send the X coordinate of the pixel</li> <li>3. Send the Y coordinate of the pixel</li> <li>4. Wait until receive a module available byte ('E') from LED Display</li> <li>5. Wait 2ms</li> </ol>	<pre>void Set_Pixel( int X0_Pos, int Y0_Pos) {   Serial.write(0x9e);   Serial.write(X0_Pos);   Serial.write(Y0_Pos);   while (Serial.read() !='E') {}   delay(2); }</pre>
0x9f	Set a pixel for negative display (clear pixel)	<ol style="list-style-type: none"> <li>1. Send 0x9f</li> <li>2. Send the X coordinate of the pixel</li> <li>3. Send the Y coordinate of the pixel</li> <li>4. Wait until receive a module available byte ('E') from LED Display</li> <li>5. Wait 2ms</li> </ol>	<pre>void Clear_Pixel( int X0_Pos, int Y0_Pos) {   Serial.write(0x9f);   Serial.write(X0_Pos);   Serial.write(Y0_Pos);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xa0	Display image row by row Up Ward	<ol style="list-style-type: none"> <li>1. Send 0xa0</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Row_By_Row_Up_Ward( int Speed) {   Serial.write(0xa0);   Serial.write(speed);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xa1	Display image row by row Down Ward	<ol style="list-style-type: none"> <li>1. Send 0xa1</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Row_By_Row_Down_Ward( int speed) {   Serial.write(0xa1);   Serial.write(speed);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xa2	Display image column by column Left Ward	<ol style="list-style-type: none"> <li>1. Send 0xa2</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Column_By_Column_Left_Ward( int speed) {   Serial.write(0xa2);   Serial.write(Speed);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xa3	Display image column by column Right Ward	<ol style="list-style-type: none"> <li>1. Send 0xa3</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Column_By_Column_Right_Ward( int Speed) {   Serial.write(0xa3);   Serial.write(Speed);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xa4	Erase image row by row Up Ward	<ol style="list-style-type: none"> <li>1. Send 0xa4</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Row_By_Row_Up_Ward( int Speed) {   Serial.write(0xa4);   Serial.write(Speed);   while (Serial.read() !='E') {}   delay(2); }</pre>



## Model No: ACM053B

0xa5	Erase image row by row Down Ward	<ol style="list-style-type: none"> <li>1. Send 0xa5</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Row_By_Row_Down_Ward( int Speed) {     Serial.write(0xa5);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xa6	Erase image column by column Left Ward	<ol style="list-style-type: none"> <li>1. Send 0xa6</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Column_By_Column_Left_Ward( int Speed) {     Serial.write(0xa6);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xa7	Erase image column by column Right Ward	<ol style="list-style-type: none"> <li>1. Send 0xa7</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Column_By_Column_Right_Ward( int Speed) {     Serial.write(0xa7);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xa8	Display image Inside Out	<ol style="list-style-type: none"> <li>1. Send 0xa8</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Inside_Out( int Speed) {     Serial.write(0xa8);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xa9	Display image Outside In	<ol style="list-style-type: none"> <li>1. Send 0xa9</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_Outside_In( int Speed) {     Serial.write(0xa9);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xaa	Erase image Inside Out	<ol style="list-style-type: none"> <li>1. Send 0xaa</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Inside_Out( int Speed) {     Serial.write(0xaa);     Serial.write(Speed);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xab	Erase image Outside In	<ol style="list-style-type: none"> <li>1. Send 0xab</li> <li>2. Send the speed (typical time is 20ms)</li> <li>3.Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Erase_Outside_In( int Speed) {     Serial.write(0xab);     Serial.write(Speed);     while (Serial.read() !='E') {} }</pre>
0xd0	Clear display	<ol style="list-style-type: none"> <li>1. Send 0xd0</li> <li>2.Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>void Clear_Display_Momery( void) {     Serial.write(0xd0);     while (Serial.read() !='E') {}     delay(2); }</pre>





## Model No: ACM053B

0xd1	Show the data in the display memory	<ol style="list-style-type: none"> <li>1. Send 0xd1</li> <li>2. Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>void Show_Display_Momery( void) {     Serial.write(0xd1);     while (Serial.read() !='E') {} }</pre>
0xd2	Scroll the whole display upward	<ol style="list-style-type: none"> <li>1. Send 0xd2</li> <li>2. Send the shift time (typical time is 70ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Scroll_Whole_Display_Memory_Up( int shift time) {     Serial.write(0xd2);     Serial.write(shift time);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xd3	Scroll the whole display downward	<ol style="list-style-type: none"> <li>1. Send 0xd3</li> <li>2. Send the shift time (typical time is 70ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Scroll_Whole_Display_Memory_Down( int shift time) {     Serial.write(0xd3);     Serial.write(shift time);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xd4	Scroll the whole display leftward	<ol style="list-style-type: none"> <li>1. Send 0xd4</li> <li>2. Send the shift time (typical time is 70ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Scroll_Whole_Display_Memory_Left( int shift time) {     Serial.write(0xd4);     Serial.write(shift time);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xd5	Scroll the whole display rightward	<ol style="list-style-type: none"> <li>1. Send 0xd5</li> <li>2. Send the shift time (typical time is 70ms)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Scroll_Whole_Display_Memory_Right( int shift time) {     Serial.write(0xd5);     Serial.write(shift time);     while (Serial.read() !='E') {}     delay(2); }</pre>
0xd6	Scroll the section display upward	<ol style="list-style-type: none"> <li>1. Send 0xd6</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send the shift time (typical time is 20ms)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Scroll_Section_Display_Memory_Up( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) {     Serial.write(0xd6);     Serial.write(X0_Pos);     Serial.write(Y0_Pos);     Serial.write(X1_Pos);     Serial.write(Y1_Pos);     Serial.write(shift time);     while (Serial.read() !='E') {};     delay(2); }</pre>



## Model No: ACM053B

0xd7	Scroll the section display downward	<ol style="list-style-type: none"> <li>1. Send 0xd7</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send the shift time (typical time is 70ms)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Scroll_Section_Display_Memory_Down( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Serial.write(0xd7); Serial.write(X0_Pos); Serial.write(Y0_Pos); Serial.write(X1_Pos); Serial.write(Y1_Pos); Serial.write(shift time); while (Serial.read() !='E') {} delay(2); }</pre>
0xd8	Scroll the section display leftward	<ol style="list-style-type: none"> <li>1. Send 0xd8</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send the shift time (typical time is 20ms)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Scroll_Section_Display_Memory_Left( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Serial.write(0xd8); Serial.write(X0_Pos); Serial.write(Y0_Pos); Serial.write(X1_Pos); Serial.write(Y1_Pos); Serial.write(shift time); while (Serial.read() !='E') {} delay(2); }</pre>
0xd9	Scroll the section display rightward	<ol style="list-style-type: none"> <li>1. Send 0xd9</li> <li>2. Send the X coordinate of up left corner</li> <li>3. Send the Y coordinate of up left corner</li> <li>4. Send the X coordinate of bottom right corner</li> <li>5. Send the Y coordinate of bottom right corner</li> <li>6. Send the shift time (typical time is 70ms)</li> <li>7. Wait until receive a module available byte ('E') from LED Display</li> <li>8. Wait 2ms</li> </ol>	<pre>void Scroll_Section_Display_Memory_Right( int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Serial.write(0xd9); Serial.write(X0_Pos); Serial.write(Y0_Pos); Serial.write(X1_Pos); Serial.write(Y1_Pos); Serial.write(shift time); while (Serial.read() !='E') {} delay(2); }</pre>
0xda	Display quarter of display memory (Available for Mode0, 1, and 2 only)	<ol style="list-style-type: none"> <li>1. Send 0xda</li> <li>2. Send the quarter No. to display</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Display_quadrant( int quadrant) { Serial.write(0xda); Serial.write(quadrant); while (Serial.read() !='E') {} delay(2); }</pre>
0xf0	Turn display Off	<ol style="list-style-type: none"> <li>1. Send 0xf0</li> <li>2. Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>void Display_Off( void) { Serial.write(0xf0); while (Serial.read() !='E') {} }</pre>
0xf1	Turn display On	<ol style="list-style-type: none"> <li>1. Send 0xf1</li> <li>2. Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>void Display_On( void) { Serial.write(0xf1); while (Serial.read() !='E') {} delay(2); }</pre>



## Model No: ACM053B

0xf2	Set the brightness of the LED Display	<ol style="list-style-type: none"> <li>1. Send 0xf2</li> <li>2. Send the level of brightness (0~11)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Set_Display_Brightness( int brightness) {   Serial.write(0xf2);   Serial.write(brightness);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xf3	Inverse image	<ol style="list-style-type: none"> <li>1. Send 0xf3</li> <li>2. Wait until receive a module available byte ('E') from LED Display</li> <li>3. Wait 2ms</li> </ol>	<pre>void Inverse_Image( void) {   Serial.write(0xf3);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xf6	Change Instruction mode (0 for Hex Command, 1 for AT Command)	<ol style="list-style-type: none"> <li>1. Send 0xf6</li> <li>2. Send instruction mode 1</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>int Change_Display_Mode(int mode) {   Serial.write(0xf6);   Serial.write(1);   while (Serial.read() !='E') {}   delay(2); }</pre>
0xf7	Change Display Mode	<ol style="list-style-type: none"> <li>1. Send 0xf7</li> <li>2. Send Display mode (0~12)</li> <li>3. Wait until receive a module available byte ('E') from LED Display</li> <li>4. Wait 2ms</li> </ol>	<pre>void Change_Display_Mode( int mode) {   Serial.write(0xf7);   Serial.write(mode);   while (Serial.read() !='E') {}   delay(2); }</pre>



**ASCII code of 5X7 fonts and 8X16 fonts**

Hex	Symbol	Hex	Symbol	Hex	Symbol
0x20		0x40	@	0x60	`
0x21	!	0x41	A	0x61	a
0x22	"	0x42	B	0x62	b
0x23	#	0x43	C	0x63	c
0x24	\$	0x44	D	0x64	d
0x25	%	0x45	E	0x65	e
0x26	&	0x46	F	0x66	f
0x27		0x47	G	0x67	g
0x28	(	0x48	H	0x68	h
0x29	)	0x49	I	0x69	i
0x2a	*	0x4a	J	0x6a	j
0x2b	+	0x4b	K	0x6b	k
0x2c	,	0x4c	L	0x6c	l
0x2d	-	0x4d	M	0x6d	m
0x2e	.	0x4e	N	0x6e	n
0x2f		0x4f	O	0x6f	o
0x30	0	0x50	P	0x70	p
0x31	1	0x51	Q	0x71	q
0x32	2	0x52	R	0x72	r
0x33	3	0x53	S	0x73	s
0x34	4	0x54	T	0x74	t
0x35	5	0x55	U	0x75	u
0x36	6	0x56	V	0x76	v
0x37	7	0x57	W	0x77	w
0x38	8	0x58	X	0x78	x
0x39	9	0x59	Y	0x79	y
0x3a	:	0x5a	Z	0x7a	z
0x3b	;	0x5b	]	0x7a	{
0x3c	<	0x5c	\	0x7a	
0x3d	=	0x5d	[	0x7a	}
0x3e	>	0x5e	^	0x7a	~
0x3f	?	0x5f	_	0x7a	<-

**ASCII code of  
16X16 fonts**

Hex	Symbol
0x30	0
0x31	1
0x32	2
0x33	3
0x34	4
0x35	5
0x36	6
0x37	7
0x38	8
0x39	9

**No. of 8X16 pattern**

No.	Symbol
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

**No. of 8X8 pattern**

No.	Symbol
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

**No. of 16X16  
pattern**

No.	Symbol
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

**No. of 32X32  
pattern**

No.	Symbol
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	°C
11	°F
12	☼



## **HANDLING PRECAUTION**

### **1. Mounting Method**

The panel of the LCD Module consists of two thin glass plates with polarizers which easily get damaged since the Module is fixed by utilizing fitting holes in the printed circuit board.

Extreme care should be taken when handling the LCD Modules.

### **2. Caution of LCD handling & cleaning**

When cleaning the display surface, use soft cloth with solvent (recommended below) and Wipe lightly.

- Isopropyl alcohol
- Ethyl alcohol
- Trichlorotrifluoroethane

Do not wipe the display surface with dry or hard materials that will damage the polarizer surface.

Do not use the following solvent :

- Water
- Kettle
- Aromatics

### **3. Caution against static charge**

The LCD Module use C-MOSLSI drivers, so we recommend that you connect any unused input terminal to VDD or VSS, do not input any signals before power is turned on. And ground your body, Work/assembly table. And assembly equipment to protect against static electricity.

### **4. Packaging**

-Modules use LCD elements, and must be treated as such. Avoid in tense shock and falls from a height.

-To prevent modules from degradation. Do not operate or store them exposed directly to sunshine or high temperature/humidity.

## 5. Caution for operation

-It is indispensable to drive LCD's with in the specified voltage limit since the higher voltage than the limit shorten LCD life.

An electrochemical reaction due to direct current causes LCD deterioration, Avoid the use of direct current drive.

-Response time will be extremely delayed at lower temperature than the operating temperature range and on the other hand at higher temperature LCD's show dark color in them. However those phenomena do not mean malfunction or out of order with LCD's. Which will come back in the specified operating temperature range.

- If the display area is pushed hard during operation, some font will be abnormally displayed but it resumes normal condition after turning off once.

- A slight dew depositing on terminals is a cause for electro-chemical reaction resulting in terminal open circuit. Usage under the relative condition of 40 °C, 50%RH or less is required.

## 6. Storage

In the case of storing for a long period of time (for instance. For years) for the purpose or replacement use, The following ways are recommended.

- Storage in a polyethylene bag with sealed so as not to enter fresh air outside in it, And with no desiccant.

- Placing in a dark place where neither exposure to direct sunlight nor light is. Keeping temperature in the specified storage temperature range.

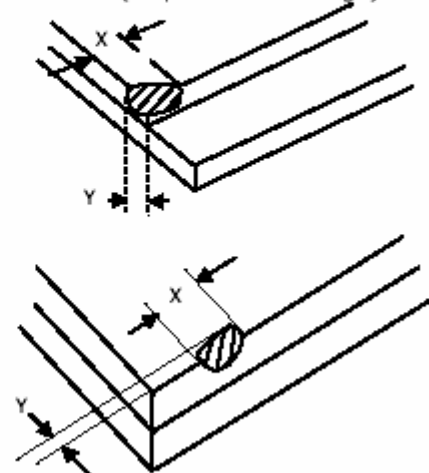
-Storing with no touch on polarizer surface by the anything else. (It is recommended to store them as they have been contained in the inner container at the time of delivery)

## 7. Safety

- It is recommendable to crash damaged or unnecessary LCD into pieces and wash off liquid crystal by using solvents such as acetone and ethanol. Which should be burned up later.

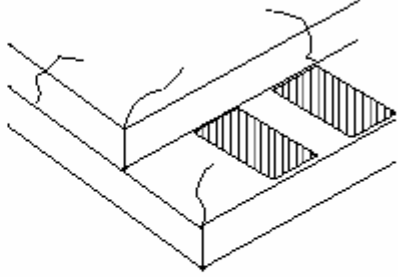

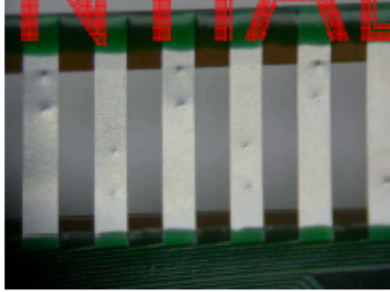
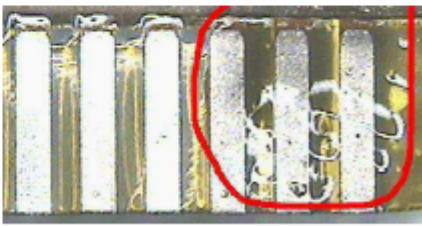
- When any liquid crystal leaked out of a damaged glass cell comes in contact with your hands, please wash it off well with soap and water.

Cosmetic Check (Display Off) in Non-Active Area

Check Item	Classification	Criteria
Panel General Chipping	Minor	<p>X &gt; 6 mm (Along with Edge)                      Y &gt; 1 mm (Perpendicular to edge)</p> 



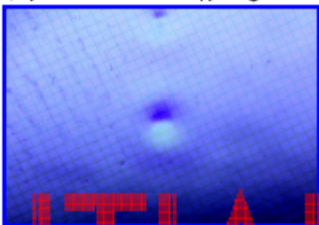
Cosmetic Check (Display Off) in Non-Active Area (Continued)

Check Item	Classification	Criteria
Panel Crack	Minor	Any crack is not allowable. 
Copper Exposed (Even Pin or Film)	Minor	Not Allowable by Naked Eye Inspection
Film or Trace Damage	Minor	
Terminal Lead Prober Mark	Acceptable	
Glue or Contamination on Pin (Couldn't Be Removed by Alcohol)	Minor	
Ink Marking on Back Side of panel (Exclude on Film)	Acceptable	Ignore for Any

CONFIDENTIAL

Cosmetic Check (Display Off) in Active Area

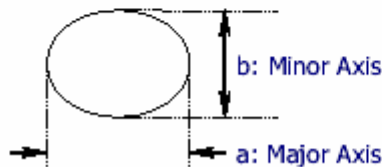
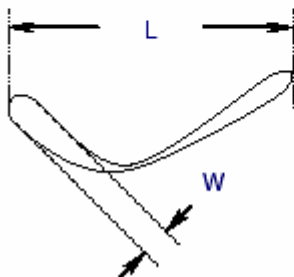
It is recommended to execute in clear room environment (class 10k) if actual in necessary.

Check Item	Classification	Criteria
Any Dirt & Scratch on Polarizer's Protective Film	Acceptable	Ignore for not Affect the Polarizer
Scratches, Fiber, Line-Shape Defect (On Polarizer)	Minor	$W \leq 0.1$ Ignore $W > 0.1$ $L \leq 2$ $n \leq 1$ $L > 2$ $n = 0$
Dirt, Black Spot, Foreign Material, (On Polarizer)	Minor	$\Phi \leq 0.1$ Ignore $0.1 < \Phi \leq 0.25$ $n \leq 1$ $0.25 < \Phi$ $n = 0$
Dent, Bubbles, White spot (Any Transparent Spot on Polarizer)	Minor	$\Phi \leq 0.5$ → Ignore if no Influence on Display $0.5 < \Phi$ $n = 0$ 
Fingerprint, Flow Mark (On Polarizer)	Minor	Not Allowable


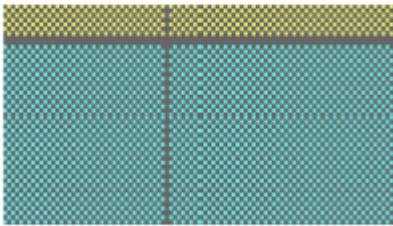
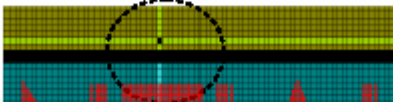
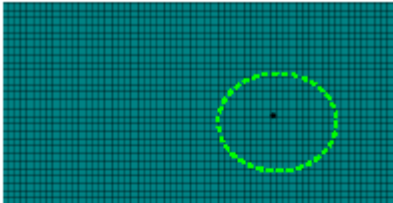
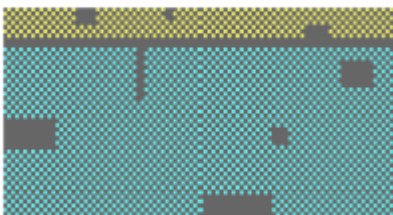

CONFIDENTIAL

\* Protective film should not be tear off when cosmetic check.

\*\* Definition of W & L &  $\Phi$  (Unit: mm):  $\Phi = (a + b) / 2$



Pattern Check (Display On) in Active Area

Check Item	Classification	Criteria
No Display	Major	
Missing Line	Major	
Pixel Short	Major	
Darker Pixel	Major	
Wrong Display	Major	
Un-uniform	Major	

CONFIDENTIAL