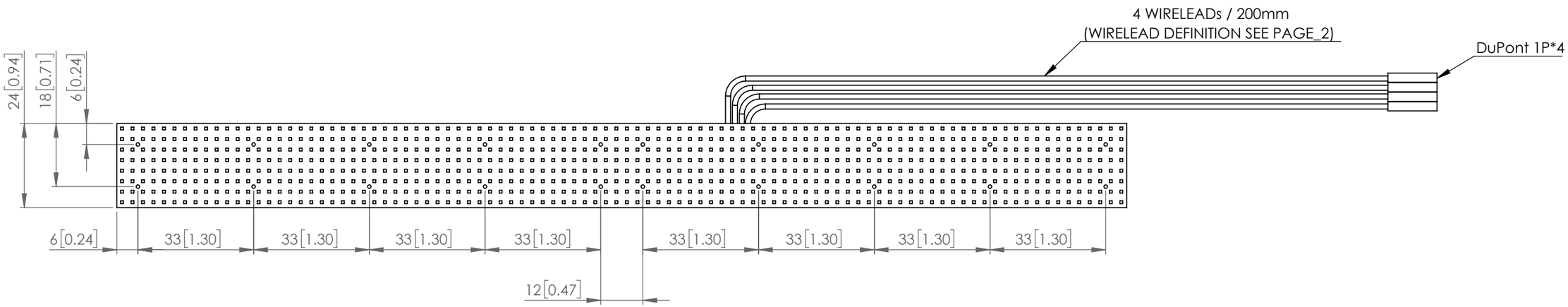
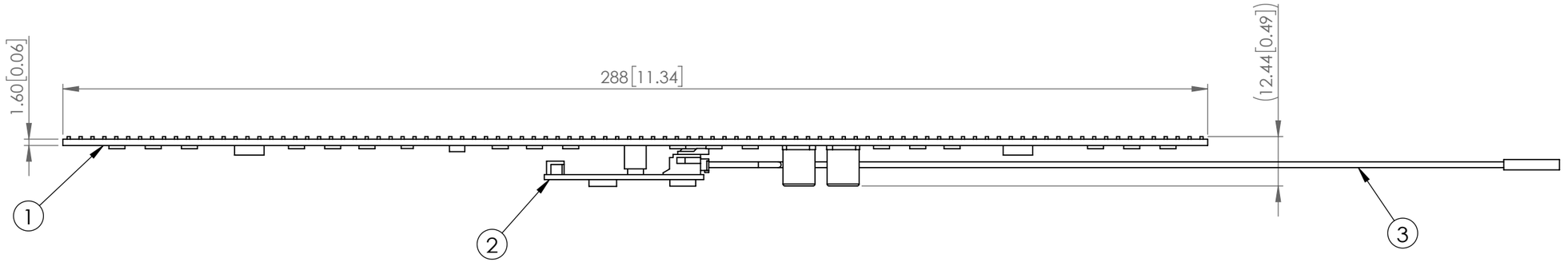
PART NUMBER	LDM-768-1LT-RGB1	REV.	--





WIRELEAD DEFINITION :

COLOR	DEFINITION
YELLOW	TX1
WHITE	RX1
RED	5V
BLACK	GND

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX®
Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

****THE SPECIFICATIONS MAY CHANGE AT ANY TIME WITHOUT NOTICE DUE TO NEW MATERIALS OR PRODUCT IMPROVEMENT.****

DATE : 2017.11.21	DRAWN BY : C.C.
PAGE : 1 OF 18	CHKD BY : E.C.
SCALE : NTF	APRVD BY : G.Y.
UNIT : mm [INCH]	Pb

BOM :

P/N	ITEM	COMPONENT	QTY
LDM-768-1LT-RGB1	1	LDM-768-1LT-RGB-PCB	1
	2	LDM-768-1LT-RGB	1
	3	WIRE002	1

UART CONFIGURATION :

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

LED ELECTRO-OPTICAL CHARACTERISTICS TA =25°C : PER LED

PARAMETER		MIN	TYP	MAX	UNITS	TEST COND
PEAK WAELENGHT	**R**	-	630	-	nm	If=3mA
	G	-	520	-		If=5mA
	B	-	460	-		If=5mA
FORWARD VOLTAGE	**R**	-	2.0	2.4	Vf	If=3mA
	G	-	3.3	3.7		If=5mA
	B	-	3.3	3.7		If=5mA
REVERSE VOLTAGE (R/GB)		-	-	5	Vr	Ir=10uA
LUMINOUS INTENSITY	**R**	7.5	-	18	mcd	If=3mA
	G	24	-	72		If=5mA
	B	6	-	19		If=5mA
VIEWING ANGLE		-	120	-	2x theta1/2	-
EMITTED COLOR		R/G/B				
EPOXY LENS FINISH		BLACK DIFFUSED				

LED LIMITS OF SAFE OPERATION AT 25°C : PER LED

PARAMETER	MAX	UNITS
PEAK FORWARD CURRENT (R/G/B)*	20/20/20	mA
FORWARD CURRENT (R/G/B)	10/10/10	mA
POWER DISIPATION (R/G/B)	24/35/35	mW
STORAGE TEMPERATURE	-40 TO +90	℃
OPERATING TEMPERATURE	-40 TO +85	℃

MAX. LOAD CURRENT & POWER CONSUMPTION WITH ALL LED ON :

PARAMETER	MAX	UNIT	TEST COND.
LOAD CURRENT	1200	mA	5V
POWER CONSUMPTION	6	W	5V

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

DATE : 2017.11.21	DRAWN BY : C.C.
PAGE : 2 OF 18	CHKD BY : E.C.
SCALE : NTF	APRVD BY : G.Y.
UNIT : mm [INCH]	Pb

ezDisplay RGD LED Display Command List

Default baud rate of LED Display is 115200 **Except for codes for 0xd1~0xd9, 0xa0~ab and 0xf0~0xf6 all other codes only change the display memory, thus you have to excute the "0xd1" code (Function of refresh the display) to display the changes in the display memory after you write a character, string , pattern or draw a line.**	**Wrong Example:** Write_5X7_String(7, 17 , positive, "RPM"); Write_8X16_Pattern(1, 45, positive, 0); Draw_Rectangle(0, 0, 127, 127, positive) ; /*Without excute the Show_Display_Momery() function the change only in the memory, it won't display */	**Correct Example:** Write_8X16_Pattern(1, 45, positive, 0); Draw_Rectangle(0, 0, 127, 127, positive) ; **Show_Display_Momery();** /*With the execution of Show_Display_Memory() fucntion, the change of display memory will be displayed*/

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

N/A	Send Image (96X8 bitmap) to LED Display
	
	(An array consist of 96 bytes bitmap)
		1. A "for" loop to send 96 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 96; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);
				1. A "for" loop to send 96 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 96; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2);

N/A	Send Image (96X8 bitmap) to LED Display
	
	(An array consist of 768 bytes bitmap)
		1. A "for" loop to send 768 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 768; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);
				1. A "for" loop to send 768 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 768; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2);

N/A	Send Image (96x32 bitmap) to LED Display
	
	(An array consist of 384 bytes bitmap)
		1. A "for" loop to send 384 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 384; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);
				1. A "for" loop to send 384 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 384; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2);

N/A	Send Image (96x32 bitmap) to LED Display
	
	(An array consist of 3072 bytes bitmap)
		1. A "for" loop to send 3072 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 3072; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);
				1. A "for" loop to send 3072 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 3072; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2);

N/A	Send Image (192x64 bitmap) to LED Display
	
	(An array consist of 1536 bytes bitmap)
		1. A "for" loop to send 1536 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 1536; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);
				1. A "for" loop to send 1536 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 1536; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2);

N/A	Send Image (192x64 bitmap) to LED Display
	
	(An array consist of 12288 bytes bitmap)
		1. A "for" loop to send 12288 bytes user define display information
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms
			for (i = 0 ; i < 12288; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2)
				1. A "for" loop to send 12288 bytes user define display information
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
					for (i = 0 ; i < 12288; i++)
					{
					Serial.write(User_define_array[i]);
					}
					while (Serial.read() !='E') {}
					delay(2)

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**
N/A	Text input without AT command	**Text input 5x7 string** Total 1 line, and 16 characters per line could be input on Mono LED 96x8 screen Total 4 lines, and 16 characters per line could be input on Mono LED 96x32 screen Total 8 lines, and 32 characters per line could be input on Mono LED 192x64 screen			

0x80	Write a 5X7 Character	1. Send **0x80**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send character's ASCII code
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_5X7_Character(int line, int column, char Char) {
			Serial.write(**0x80**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Char);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**80**=(line,column, Character)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT80=(0,0,A)
					Write_AT_Command("AT**80**=(0,0,A)");
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x81	Write a 5X7 String	1. Send **0x81**
		2. Send which line to start the string
		3. Send which cloumn to start the string
		4. Send string
		5. Wait until receive a module available byte('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_5X7_String(int line, int column, char * string) {
			Serial.write(**0x81**);
			Serial.write(line);
			Serial.write(column);
			Serial.print(string);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**81**=(line,column,String)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT81=(0,0,ABCD1234)
					Write_AT_Command("AT**81**=(0,0,ABCD1234)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x82	Write a 8X16 Character
	
	(not available for 96x8 RGB LED)
		1. Send **0x82**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send character's ASCII code
		5.Wait until receive a module available byte('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_8X16_Character(int line, int column, char Char) {
			Serial.write(**0x82**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Char);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**82**=(line,column,Character)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT82=(0,0,A)
					Write_AT_Command("AT**82**=(0,0,A)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x83	Write a 8X16 String
	
	(not available for 96x8 RGB LED)
		1. Send **0x83**
		2. Send which line to stary the string
		3. Send which cloumn to start the string
		4. Send string
		5. Wait until receive a module available byte('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_8X16_String(int line, int column, char * string) {
			Serial.write(**0x83**);
			Serial.write(line);
			Serial.write(column);
			Serial.print(string);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**83**=(line,column,String)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT83=(0,0,ABCD1234)
					Write_AT_Command("AT**83**=(0,0,ABCD1234)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x84	Dsiplay a 8X8 pattern	1. Send **0x84**
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		4. Send the ID of pattern
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_8X8_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, 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);
			}
				1. AT**84**=(X position, Y position, pattern ID)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT84=(16,32,1)
					Write_AT_Command("AT**84**=(16,32,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0x85	Dsiplay a 8X16 pattern
	
	(not available for 96x8 RGB LED)
		1. Send **0x85**
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		4. Send the ID of pattern
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_8X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, 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);
			}
				1. AT**85**=(X position,Y position,pattern ID)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT85=(16,32,1)
					Write_AT_Command("AT**85**=(16,32,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x86	Dsiplay a 16X16 pattern
	
	(not available for 96x8 RGB LED)
		1. Send **0x86**
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		4. Send the ID of pattern
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_16X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, 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);
			}
				1. AT**86**=(X position,Y position,pattern ID)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT86=(16,32,1)
					Write_AT_Command("AT**86**=(16,32,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x87	Dsiplay a 32X32 pattern
	
	(not available for 96x8 RGB LED)
		1. Send **0x87**
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		4. Send the ID of pattern
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED.
			void Write_32X32_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, 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);
			}
				1. AT**87**=(X position,Y position,pattern ID)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT87=(16,32,1)
					Write_AT_Command("AT**87**=(16,32,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x88	Dsiplay a 16X32 pattern
	
	(not available for 96x8 RGB LED)
		1. Send **0x88**
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		4. Send the ID of pattern
		5. Wait until receive a module available byte ('E') from ezDisplay
		6. Wait 2ms, but NO need for OLED
			void Write_16X32_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int Pattern_ID) {
			Serial.write(**0x88**);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**88**=(X position,Y position,pattern ID)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				AT88=(16,32,1)
					Write_AT_Command("AT**88**=(16,32,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x90	Draw a line	1. Send **0x90**
		2. Send the X coordinate of first point
		3. Send the Y coordinate of first point
		4. Send the X coordinate of second point
		5. Send the Y coordinate of second point
		6. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			void Draw_Line(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int positive) {
			Serial.write(**0x90**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**90**=(X0 position,Y0 position,X1 position,Y1 position,**0~255**)
				
				*0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				AT90=(1,4,94,4,4) : '4' green color
					Write_AT_Command("AT**90**=(1,4,94,4,4)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= +DECIMAL PRECISION -0.00 MAX.= +0.00 -DECIMAL PRECISION

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0x91	Draw a Rectangle	1. Send **0x91**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			void Draw_Rectangle(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int positive) {
			Serial.write(**0x91**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**91**=(X0 position,Y0 position,X1 position,Y1 position,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT91=(0,0,95,7,2) : '2' blue color
					Write_AT_Command("AT**91**=(0,0,95,7,2)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x92	Draw a filled Rectangle	1. Send **0x92**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			void Draw_Filled_Rectangle(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int positive) {
			Serial.write(**0x92**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**92**=(X0 position,Y0 position,X1 position,Y1 position,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT92=(1,1,94,6,32) : '32' red color
					Write_AT_Command("AT**92**=(1,1,94,6,32)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x93	Draw a Square	1. Send **0x93**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the width of this square
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Square(int X0_Pos, int Y0_Pos, int width, int positive) {
			Serial.write(**0x93**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**93**=(X position,Y position,Width,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT93=(48,1,4,2) : '2' blue color
					Write_AT_Command("AT**93**=(48,1,4,2)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x94	Draw a Circle	1. Send **0x94**
		2. Send the X coordinate of the center
		3. Send the Y coordinate of the center
		4. Send the radius of this circle
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Circle(int X0_Pos, int Y0_Pos, int radius, int positive) {
			Serial.write(**0x94**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(radius);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**94**=(X position,Y position,Radius,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT94=(48,4,3,2) : '2' blue color
					Write_AT_Command("AT**94**=(48,4,3,2)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x95	Draw a filled Circle	1. Send **0x95**
		2. Send the X coordinate of the center
		3. Send the Y coordinate of the center
		4. Send the radius of this circle
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Filled_Circle(int X0_Pos, int Y0_Pos, int radius, int positive) {
			Serial.write(**0x95**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(radius);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**95**=(X position,Y position,Radius,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT95=(48,4,3,4) : '4' green color
					Write_AT_Command("AT**95**=(48,4,3,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX®
Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

DATE : 2017.11.21	DRAWN BY : C.C.
PAGE : 6 OF 18	CHKD BY : E.C.
SCALE : NTF	APRVD BY : G.Y.
UNIT : mm [INCH]	Pb

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0x96	Draw a tip upward Triangle	1. Send **0x96**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the height of the tip to the bottom
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Triangle_Up_Ward(int X0_Pos, int Y0_Pos, int height, int positive) {
			Serial.write(**0x96**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**96**=(X position,Y position,Height,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT96=(48,1,3,4) : '4' green color
					Write_AT_Command("AT**96**=(48,1,3,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x97	Draw a filled tip upward Triangle	1. Send **0x97**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the height of the tip to the bottom
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Filled_Triangle_Up_Ward(int X0_Pos, int Y0_Pos, int height, int positive) {
			Serial.write(**0x97**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**97**=(X position,Y position,Height,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT97=(48,1,3,32) : '32' red color
					Write_AT_Command("AT**97**=(48,1,3,32)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x98	Draw a tip downward Triangle	1. Send **0x98**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the height of the tip to the top
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Triangle_Down_Ward(int X0_Pos, int Y0_Pos, int height, int positive) {
			Serial.write(**0x98**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**98**=(X position,Y position,Height,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT98=(48,6,3,4) : '4' green color
					Write_AT_Command("AT**98**=(48,6,3,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x99	Draw a filled tip downward Triangle	1. Send **0x99**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the height of the tip to the top
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw__Filled_Triangle_Down_Ward(int X0_Pos, int Y0_Pos, int height, int positive) {
			Serial.write(**0x99**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**99**=(X position,Y position,Height,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT99=(48,6,3,4) : '4' green color
					Write_AT_Command("AT**99**=(48,6,3,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x9a	Draw a tip leftward Triangle	1. Send **0x9a**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the width of the tip to the right
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int positive) {
			Serial.write(**0x9a**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**9a**=(X position,Y position,Width,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT9a=(48,4,2,4) : '4' green color
					Write_AT_Command("AT**9a**=(48,4,2,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0x9b	Draw a filled tip leftward Triangle	1. Send **0x9b**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the width of the tip to the right
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Filled_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int positive) {
			Serial.write(**0x9b**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**9b**=(X position,Y position,Width,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT9b=(48,4,2,4) : '4' green color
					Write_AT_Command("AT**9b**=(48,4,2,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x9c	Draw a tip rightward Triangle	1. Send **0x9c**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the width of the tip to the left
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int positive) {
			Serial.write(**0x9c**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**9c**=(X position,Y position,Width, **0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT9c=(48,4,2,4) : '4' green color
					Write_AT_Command("AT**9c**=(48,4,2,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x9d	Draw a filled tip rightward Triangle	1. Send **0x9d**
		2. Send the X coordinate of the tip
		3. Send the Y coordinate of the tip
		4. Send the width of the tip to the left
		5. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Draw_Filled_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int positive) {
			Serial.write(**0x9d**);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(positive);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**9d**=(X position,Y position,Width,**0~255**)
				*0~255: ref to **COLOR CODE Table**
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<RGB LED example>
				AT9d=(48,4,2,4) : '4' green color
					Write_AT_Command("AT**9d**=(48,4,2,4)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x9e	Set a pixel for positive display (show pixel)	1. Send **0x9e**
		2. Send the X coordinate of the pixel
		3. Send the Y coordinate of the pixel
		4. Wait until receive a module available byte ('E') from ezDisplay
		5. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**9e**=(X position,Y position)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<example>
				AT9e=(120,32)
					Write_AT_Command("AT**9e**=(120,32)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0x9f	Set a pixel for negative display (clear pixel)	1. Send **0x9f**
		2. Send the X coordinate of the pixel
		3. Send the Y coordinate of the pixel
		4. Wait until receive a module available byte ('E') from ezDisplay
		5. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**9f**=(X position,Y position)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				<example>
				AT9f=(120,32)
					Write_AT_Command("AT**9f**=(120,32)")
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xa0	Display image row by row Up Ward	1. Send **0xa0**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Row_By_Row_Up_Ward(int Speed) {
			Serial.write(**0xa0**);
			Serial.write(speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a0**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa0=(20)
					Write_AT_Command("AT**a0**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa1	Display image row by row Down Ward	1. Send **0xa1**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Row_By_Row_Down_Ward(int speed) {
			Serial.write(**0xa1**);
			Serial.write(speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a1**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa1=(20)
					Write_AT_Command("AT**a1**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa2	Display image column by column Left Ward	1. Send **0xa2**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Column_By_Column_Left_Ward(int speed) {
			Serial.write(**0xa2**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a2**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa2=(20)
					Write_AT_Command("AT**a2**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa3	Display image column by column Right Ward	1. Send **0xa3**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Column_By_Column_Right_Ward(int Speed) {
			Serial.write(**0xa3**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a3**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa3=(20)
					Write_AT_Command("AT**a3**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa4	Erase image row by row Up Ward	1. Send **0xa4**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Row_By_Row_Up_Ward(int Speed) {
			Serial.write(**0xa4**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a4**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa4=(20)
					Write_AT_Command("AT**a4**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xa5	Erase image row by row Down Ward	1. Send **0xa5**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Row_By_Row_Down_Ward(int Speed) {
			Serial.write(**0xa5**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a5**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa5=(20)
					Write_AT_Command("AT**a5**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa6	Erase image column by column Left Ward	1. Send **0xa6**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Column_By_Column_Left_Ward(int Speed) {
			Serial.write(**0xa6**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a6**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa6=(20)
					Write_AT_Command("AT**a6**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa7	Erase image column by column Right Ward	1. Send **0xa7**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Column_By_Column_Right_Ward(int Speed) {
			Serial.write(**0xa7**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a7**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa7=(20)
					Write_AT_Command("AT**a7**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa8	Display image Inside Out	1. Send **0xa8**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Inside_Out(int Speed) {
			Serial.write(**0xa8**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a8**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa8=(20)
					Write_AT_Command("AT**a8**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xa9	Display image Outside In	1. Send **0xa9**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Display_Outside_In(int Speed) {
			Serial.write(**0xa9**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**a9**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATa9=(20)
					Write_AT_Command("AT**a9**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xaa	Erase image Inside Out	1. Send **0xaa**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Inside_Out(int Speed) {
			Serial.write(**0xaa**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**aa**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATaa=(20)
					Write_AT_Command("AT**aa**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xab	Erase image Outside In	1. Send **0xab**
		2. Send the speed (typical time is 20ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Erase_Outside_In(int Speed) {
			Serial.write(**0xab**);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**ab**=(Speed in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATab=(20)
					Write_AT_Command("AT**ab**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd0	Clear display	1. Send **0xd0**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Clear_Display_Momery(void) {
			Serial.write(**0xd0**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d0**=()
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd0=()
					Write_AT_Command("AT**d0**=()")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd1	Show the data in the display memory	1. Send **0xd1**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Show_Display_Momery(void) {
			Serial.write(**0xd1**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d1**=()
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd1=()
					Write_AT_Command("AT**d1**=()")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd2	Scroll the whole display upward	1. Send **0xd2**
		2. Send the shift time (typical time is 70ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Scroll_Whole_Display_Memory_Up(int shift time) {
			Serial.write(**0xd2**);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d2**=(shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd2=(20)
					Write_AT_Command("AT**d2**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd3	Scroll the whole display downward	1. Send **0xd3**
		2. Send the shift time (typical time is 70ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Scroll_Whole_Display_Memory_Down(int shift time) {
			Serial.write(**0xd3**);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d3**=(shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd3=(20)
					Write_AT_Command("AT**d3**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xd4	Scroll the whole display leftward	1. Send **0xd4**
		2. Send the shift time (typical time is 70ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Scroll_Whole_Display_Memory_Left(int shift time) {
			Serial.write(**0xd4**);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d4**=(shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd4=(20)
					Write_AT_Command("AT**d4**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd5	Scroll the whole display rightward	1. Send **0xd5**
		2. Send the shift time (typical time is 70ms)
		3. Wait until receive a module available byte ('E') from ezDisplay
		4. Wait 2ms, but NO need for OLED.
			void Scroll_Whole_Display_Memory_Right(int shift time) {
			Serial.write(**0xd5**);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**d5**=(shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd5=(20)
					Write_AT_Command("AT**d5**=(20)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd6	Scroll the section display upward	1. Send **0xd6**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send the shift time (typical time is 20ms)
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**d6**=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd6=(10,16,120,50,1)
					Write_AT_Command("AT**d6**=(10,16,120,50,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd7	Scroll the section display downward	1. Send **0xd7**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send the shift time (typical time is 70ms)
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**d7**=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd7=(10,16,120,50,1)
					Write_AT_Command("AT**d7**=(10,16,120,50,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xd8	Scroll the section display leftward	1. Send **0xd8**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send the shift time (typical time is 20ms)
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**d8**=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd8=(10,16,120,50,1)
					Write_AT_Command("AT**d8**=(10,16,120,50,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xd9	Scroll the section display rightward	1. Send **0xd9**
		2. Send the X coordinate of up left corner
		3. Send the Y coordinate of up left corner
		4. Send the X coordinate of bottom right corner
		5. Send the Y coordinate of bottom right corner
		6. Send the shift time (typical time is 70ms)
		7. Wait until receive a module available byte ('E') from ezDisplay
		8. Wait 2ms, but NO need for OLED.
			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);
			}
				1. AT**d9**=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATd9=(10,16,120,50,1)
					Write_AT_Command("AT**d9**=(10,16,120,50,1)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xe0	Write a 5X7 RGB Character	1. Send **0xe0**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send 0~255 for RGB LED.
		ref to COLOR CODE Table
		5. Send character's ASCII code
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Write_5X7_RGB_Character(int line, int column, int Color, char Char) {
			Serial.write(**0xe0**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Color); **/* Send 0~255 */**
			Serial.write(Char);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**e0**=(line,column, **Color 0~255**, Character)
				*Color 0~255:ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATe0=(0,0,3,A) : '3' blue color
					Write_AT_Command("AT**e0**=(0,0,3,A)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xe1	Write a 5x7 RGB String	1. Send **0xe1**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		5. Send string
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Write_5X7_RGB_String(int line, int column, int Color, char* string) {
			Serial.write(**0xe1**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Color); **/* Send 0~255 */**
			Serial.print(string);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**e1**=(line,column, **Color 0~255**,String)
				*Color 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATe1=(0,0,3,ABCD1234) : '3' blue color
					Write_AT_Command("AT**e1**=(0,0,3,ABCD1234)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xe2	Write a 8X16 RGB Character
	
	(not available for 96x8 RGB LED)
		1. Send **0xe2**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		5. Send character's ASCII code
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Write_8x16_RGB_Character(int line, int column, int Color, char Char) {
			Serial.write(**0xe2**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Color); **/* Send 0~255 */**
			Serial.write(Char);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**e2**=(line,column, **Color 0~255**, Character)
				*Color 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATe2=(0,0,3,A) : '3' blue color
					Write_AT_Command("AT**e2**=(0,0,3,A)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xe3	Write a 8X16 RGB String
	
	(not available for 96x8 RGB LED)
		1. Send **0xe3**
		2. Send which line to put this character
		3. Send which cloumn to put this character
		4. Send **0~255** for RGB LED.
		ref to **COLOR CODE Table**
		5. Send string
		6. Wait until receive a module available byte ('E') from ezDisplay
		7. Wait 2ms, but NO need for OLED.
			void Write_8X16_RGB_String(int line, int column, int Color, char* string) {
			Serial.write(**0xe3**);
			Serial.write(line);
			Serial.write(column);
			Serial.write(Color); **/* Send 0~255 */**
			Serial.print(string);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**e3**=(line,column, **Color 0~255**,String)
				*Color 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATe3=(0,0,3,ABCD1234) : '3' blue color
					Write_AT_Command("AT**e3**=(0,0,3,ABCD1234)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

LUMEX® Creating LED and LCD Solutions Together™

N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xec	Set default Background	1. Send **0xec**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Display_Off(void){
			Serial.write(**0xec**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**ec**=(**Backgroung 0~255**)
				*Backgroung 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATec=(11) : '11' cyan color
					Write_AT_Command("AT**ec**=(11)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xee	Set a Color Pixel	1. Send **0xee**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Display_Off(void){
			Serial.write(**0xee**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**ee**=(X position,Y position, **Color 0~255**)
				*Color 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<RGB LED example>
				ATee=(48,6,96) : '96' Red color
					Write_AT_Command("AT**ee**=(48,6,96)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xef	Set default Color	1. Send **0xef**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Display_Off(void){
			Serial.write(**0xef**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**ef**=(**Color 0~255**)
				*Color 0~255: ref to **COLOR CODE Table**
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATef=(32) : '32' red color
					Write_AT_Command("AT**ef**=(32)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xf0	Turn display Off	1. Send **0xf0**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Display_Off(void){
			Serial.write(**0xf0**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**f0**=()
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATf0=()
					Write_AT_Command("AT**f0**=()")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xf1	Turn display On	1. Send **0xf1**
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			void Display_On(void) {
			Serial.write(**0xf1**);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**f1**=()
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATf1=()
					Write_AT_Command("AT**f1**=()")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

Code	Function	LED default for AT Command			
		Sequence of HEX mode	**API for Arduino (Hex mode)**	**Instruction of AT mode**	**API for Arduino (AT mode)**

0xf2	Set the brightness of RGB LED Display	1. Send **0xf2**
		2. Send level of brightness 3
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			int Set_Brightness_LED_Display(int mode) {
			Serial.write(**0xf2**);
			Serial.write(3); **/* RGD LED evel of brightness 0~3 */**
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**f2**=(**level of brightness 0~3**)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATf2=(3)
					Write_AT_Command("AT**f2**=(3)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

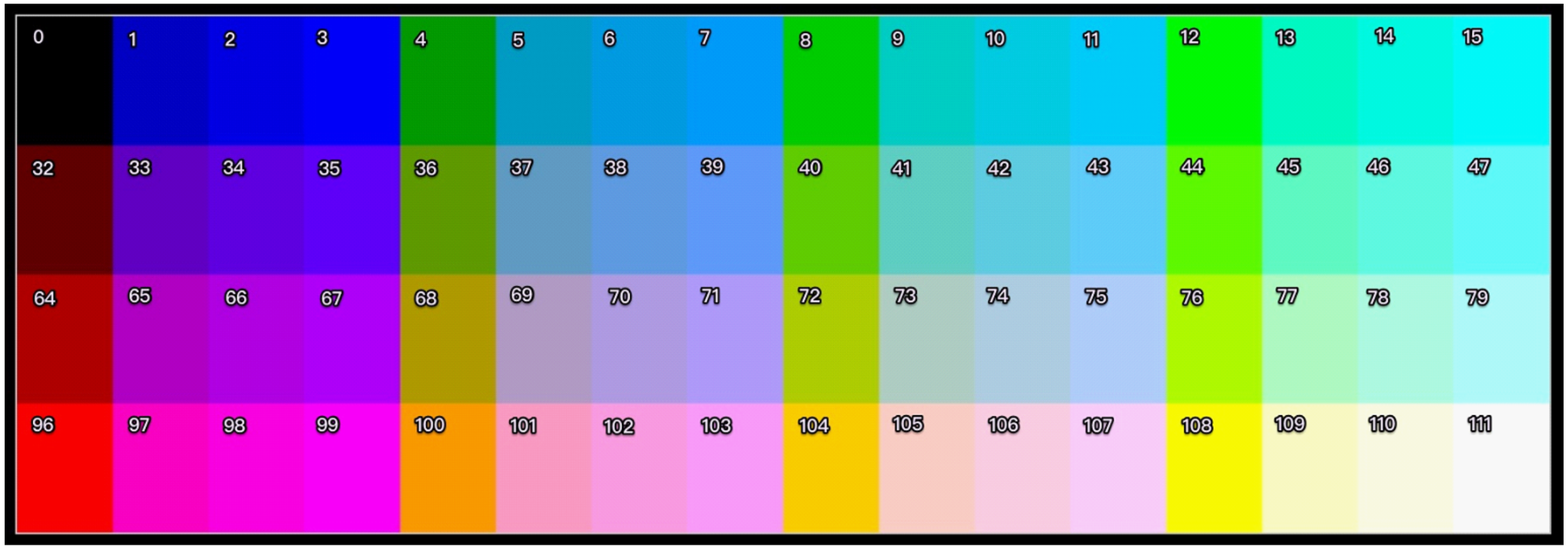
0xf6	Change Instruction mode (
	
	0 for HEX command,
	1 for AT command
	
	)
		1. Send **0xf6**
		2. Send instruction mode 1
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			int Change_Instruction_Mode(int mode) {
			Serial.write(**0xf6**);
			Serial.write(1);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**f6**=(instruction mode)
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATf6=(0)
					Write_AT_Command("AT**f6**=(0)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

0xf7	Change Display Mode	1. Send **0xf7**
		2. Send Display Mode 0 (M0 mode)
		2. Wait until receive a module available byte ('E') from ezDisplay
		3. Wait 2ms, but NO need for OLED.
			int Change_Display_Mode(int mode) {
			Serial.write(**0xf7**);
			Serial.write(0);
			while (Serial.read() !='E') {}
			delay(2);
			}
				1. AT**f7**=(Display Mode)
				*Display Mode: ref to
				「**768 LED module Configuration Mode**」
				
				2. Wait until receive a module available byte ('E') from ezDisplay
				3. Wait 2ms
				
				<example>
				ATf7=(0) : Change to M0 mode.
					Write_AT_Command("AT**f7**=(0)")
					
					void Write_AT_Command(char *string)
					{
					Serial.print(string);
					while (Serial.read() != 'E') {}
					delay(2);
					}

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

COLOR CODE Table



*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX®
Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

****THE SPECIFICATIONS MAY CHANGE AT ANY TIME WITHOUT NOTICE DUE TO NEW MATERIALS OR PRODUCT IMPROVEMENT.****

DATE : 2017.11.21	DRAWN BY : C.C.
PAGE : 16 OF 18	CHKD BY : E.C.
SCALE : NTF	APRVD BY : G.Y.
UNIT : mm [INCH]	Pb

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 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	℃
11	°F
12	

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

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= $^{+\text{DECIMAL PRECISION}}_{-0.00}$ MAX.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

PART NUMBER	LDM-768-1LT-RGB1	REV.	--

768 LED module Configuration Mode

M0 (96X32)

1	M000
2	M001
3	M002
3	M003

M1 (192X16)

1	M100	3	M102
2	M101	4	M103

M2 (384X8)

1	M200	2	M201	3	M202	4	M203

M3 (96X128)

1	M300
1	M301
1	M302
1	M303
2	M304
2	M305
2	M306
2	M307
3	M308
3	M309
3	M310
3	M311
4	M312
4	M313
4	M314
4	M315

M4 (192X64)

1	M400	3	M408
1	M401	3	M409
1	M402	3	M410
1	M403	3	M411
2	M404	4	M412
2	M405	4	M413
2	M406	4	M414
2	M407	4	M415

M5 (384X32)

1	M500	2	M504	3	M508	4	M512
1	M501	2	M505	3	M509	4	M513
1	M502	2	M506	3	M510	4	M514
1	M503	2	M507	3	M511	4	M515

M6 (96X512)

1	M600
1	M601
1	M602
1	M603
1	M604
1	M605
1	M606
1	M607
1	M608
1	M609
1	M610
1	M611
1	M612
1	M613
1	M614
1	M615
2	M616
2	M617
2	M618
2	M619
2	M620
2	M621
2	M622
2	M623
2	M624
2	M625
2	M626
2	M627
2	M628
2	M629
2	M630
2	M631
3	M632
3	M633
3	M634
3	M635
3	M636
3	M637
3	M638
3	M639
3	M640
3	M641
3	M642
3	M643
3	M644
3	M645
3	M646
3	M647
4	M648
4	M649
4	M650
4	M651
4	M652
4	M653
4	M654
4	M655
4	M656
4	M657
4	M658
4	M659
4	M660
4	M661
4	M662
4	M663

M7 (192X256)

1	M700	3	M732
1	M701	3	M733
1	M702	3	M734
1	M703	3	M735
1	M704	3	M736
1	M705	3	M737
1	M706	3	M738
1	M707	3	M739
1	M708	3	M740
1	M709	3	M741
1	M710	3	M742
1	M711	3	M743
1	M712	3	M744
1	M713	3	M745
1	M714	3	M746
1	M715	3	M747
2	M716	4	M748
2	M717	4	M749
2	M718	4	M750
2	M719	4	M751
2	M720	4	M752
2	M721	4	M753
2	M722	4	M754
2	M723	4	M755
2	M724	4	M756
2	M725	4	M757
2	M726	4	M758
2	M727	4	M759
2	M728	4	M760
2	M729	4	M761
2	M730	4	M762
2	M731	4	M763

M8 (384X128)

1	M800	2	M816	3	M832	4	M848
1	M801	2	M817	3	M833	4	M849
1	M802	2	M818	3	M834	4	M850
1	M803	2	M819	3	M835	4	M851
1	M804	2	M820	3	M836	4	M852
1	M805	2	M821	3	M837	4	M853
1	M806	2	M822	3	M838	4	M854
1	M807	2	M823	3	M839	4	M855
1	M808	2	M824	3	M840	4	M856
1	M809	2	M825	3	M841	4	M857
1	M810	2	M826	3	M842	4	M858
1	M811	2	M827	3	M843	4	M859
1	M812	2	M828	3	M844	4	M860
1	M813	2	M829	3	M845	4	M861
1	M814	2	M830	3	M846	4	M862
1	M815	2	M831	3	M847	4	M863

M9 (768X64)

1	M900	1	M908	2	M916	2	M924	3	M932	3	M940	4	M948	4	M956
1	M901	1	M909	2	M917	2	M925	3	M933	3	M941	4	M949	4	M957
1	M902	1	M910	2	M918	2	M926	3	M934	3	M942	4	M950	4	M958
1	M903	1	M911	2	M919	2	M927	3	M935	3	M943	4	M951	4	M959
1	M904	1	M912	2	M920	2	M928	3	M936	3	M944	4	M952	4	M960
1	M905	1	M913	2	M921	2	M929	3	M937	3	M945	4	M953	4	M961
1	M906	1	M914	2	M922	2	M930	3	M938	3	M946	4	M954	4	M962
1	M907	1	M915	2	M923	2	M931	3	M939	3	M947	4	M955	4	M963

M10 (1536X32)

1	M1000	1	M1004	1	M1008	1	M1012	2	M1016	2	M1020	2	M1024	2	M1028	3	M1032	3	M1036	3	M1040	3	M1044	4	M1048	4	M1052	4	M1056	4	M1060
1	M1001	1	M1005	1	M1009	1	M1013	2	M1017	2	M1021	2	M1025	2	M1029	3	M1033	3	M1037	3	M1041	3	M1045	4	M1049	4	M1053	4	M1057	4	M1061
1	M1002	1	M1006	1	M1010	1	M1014	2	M1018	2	M1022	2	M1026	2	M1030	3	M1034	3	M1038	3	M1042	3	M1046	4	M1050	4	M1054	4	M1058	4	M1062
1	M1003	1	M1007	1	M1011	1	M1015	2	M1019	2	M1023	2	M1027	2	M1031	3	M1035	3	M1039	3	M1043	3	M1047	4	M1051	4	M1055	4	M1059	4	M1063

M11 (768X256)

1	M11000	1	M11016	1	M11032	1	M11048	3	M11128	3	M11144	3	M11160	3	M11176
1	M11001	1	M11017	1	M11033	1	M11049	3	M11129	3	M11145	3	M11161	3	M11177
1	M11002	1	M11018	1	M11034	1	M11050	3	M11130	3	M11146	3	M11162	3	M11178
1	M11003	1	M11019	1	M11035	1	M11051	3	M11131	3	M11147	3	M11163	3	M11179
1	M11004	1	M11020	1	M11036	1	M11052	3	M11132	3	M11148	3	M11164	3	M11180
1	M11005	1	M11021	1	M11037	1	M11053	3	M11133	3	M11149	3	M11165	3	M11181
1	M11006	1	M11022	1	M11038	1	M11054	3	M11134	3	M11150	3	M11166	3	M11182
1	M11007	1	M11023	1	M11039	1	M11055	3	M11135	3	M11151	3	M11167	3	M11183
1	M11008	1	M11024	1	M11040	1	M11056	3	M11136	3	M11152	3	M11168	3	M11184
1	M11009	1	M11025	1	M11041	1	M11057	3	M11137	3	M11153	3	M11169	3	M11185
1	M11010	1	M11026	1	M11042	1	M11058	3	M11138	3	M11154	3	M11170	3	M11186
1	M11011	1	M11027	1	M11043	1	M11059	3	M11139	3	M11155	3	M11171	3	M11187
1	M11012	1	M11028	1	M11044	1	M11060	3	M11140	3	M11156	3	M11172	3	M11188
1	M11013	1	M11029	1	M11045	1	M11061	3	M11141	3	M11157	3	M11173	3	M11189
1	M11014	1	M11030	1	M11046	1	M11062	3	M11142	3	M11158	3	M11174	3	M11190
1	M11015	1	M11031	1	M11047	1	M11063	3	M11143	3	M11159	3	M11175	3	M11191
2	M11064	2	M11080	2	M11096	2	M11112	4	M11192	4	M11208	4	M11224	4	M11240
2	M11065	2	M11081	2	M11097	2	M11113	4	M11193	4	M11209	4	M11225	4	M11241
2	M11066	2	M11082	2	M11098	2	M11114	4	M11194	4	M11210	4	M11226	4	M11242
2	M11067	2	M11083	2	M11099	2	M11115	4	M11195	4	M11211	4	M11227	4	M11243
2	M11068	2	M11084	2	M11100	2	M11116	4	M11196	4	M11212	4	M11228	4	M11244
2	M11069	2	M11085	2	M11101	2	M11117	4	M11197	4	M11213	4	M11229	4	M11245
2	M11070	2	M11086	2	M11102	2	M11118	4	M11198	4	M11214	4	M11230	4	M11246
2	M11071	2	M11087	2	M11103	2	M11119	4	M11199	4	M11215	4	M11231	4	M11247
2	M11072	2	M11088	2	M11104	2	M11120	4	M11200	4	M11216	4	M11232	4	M11248
2	M11073	2	M11089	2	M11105	2	M11121	4	M11201	4	M11217	4	M11233	4	M11249
2	M11074	2	M11090	2	M11106	2	M11122	4	M11202	4	M11218	4	M11234	4	M11250
2	M11075	2	M11091	2	M11107	2	M11123	4	M11203	4	M11219	4	M11235	4	M11251
2	M11076	2	M11092	2	M11108	2	M11124	4	M11204	4	M11220	4	M11236	4	M11252
2	M11077	2	M11093	2	M11109	2	M11125	4	M11205	4	M11221	4	M11237	4	M11253
2	M11078	2	M11094	2	M11110	2	M11126	4	M11206	4	M11222	4	M11238	4	M11254
2	M11079	2	M11095	2	M11111	2	M11127	4	M11207	4	M11223	4	M11239	4	M11255

M12 (1536X128)

1	M11000	1	M11016	1	M11032	1	M11048	2	M11064	2	M11080	2	M11096	2	M11112	3	M11128	3	M11144	3	M11160	3	M11176	4	M11192	4	M11208	4	M11224	4	M11240
1	M11001	1	M11017	1	M11033	1	M11049	2	M11065	2	M11081	2	M11097	2	M11113	3	M11129	3	M11145	3	M11161	3	M11177	4	M11193	4	M11209	4	M11225	4	M11241
1	M11002	1	M11018	1	M11034	1	M11050	2	M11066	2	M11082	2	M11098	2	M11114	3	M11130	3	M11146	3	M11162	3	M11178	4	M11194	4	M11210	4	M11226	4	M11242
1	M11003	1	M11019	1	M11035	1	M11051	2	M11067	2	M11083	2	M11099	2	M11115	3	M11131	3	M11147	3	M11163	3	M11179	4	M11195	4	M11211	4	M11227	4	M11243
1	M11004	1	M11020	1	M11036	1	M11052	2	M11068	2	M11084	2	M11100	2	M11116	3	M11132	3	M11148	3	M11164	3	M11180	4	M11196	4	M11212	4	M11228	4	M11244
1	M11005	1	M11021	1	M11037	1	M11053	2	M11069	2	M11085	2	M11101	2	M11117	3	M11133	3	M11149	3	M11165	3	M11181	4	M11197	4	M11213	4	M11229	4	M11245
1	M11006	1	M11022	1	M11038	1	M11054	2	M11070	2	M11086	2	M11102	2	M11118	3	M11134	3	M11150	3	M11166	3	M11182	4	M11198	4	M11214	4	M11230	4	M11246
1	M11007	1	M11023	1	M11039	1	M11055	2	M11071	2	M11087	2	M11103	2	M11119	3	M11135	3	M11151	3	M11167	3	M11183	4	M11199	4	M11215	4	M11231	4	M11247
1	M11008	1	M11024	1	M11040	1	M11056	2	M11072	2	M11088	2	M11104	2	M11120	3	M11136	3	M11152	3	M11168	3	M11184	4	M11200	4	M11216	4	M11232	4	M11248
1	M11009	1	M11025	1	M11041	1	M11057	2	M11073	2	M11089	2	M11105	2	M11121	3	M11137	3	M11153	3	M11169	3	M11185	4	M11201	4	M11217	4	M11233	4	M11249
1	M11010	1	M11026	1	M11042	1	M11058	2	M11074	2	M11090	2	M11106	2	M11122	3	M11138	3	M11154	3	M11170	3	M11186	4	M11202	4	M11218	4	M11234	4	M11250
1	M11011	1	M11027	1	M11043	1	M11059	2	M11075	2	M11091	2	M11107	2	M11123	3	M11139	3	M11155	3	M11171	3	M11187	4	M11203	4	M11219	4	M11235	4	M11251
1	M11012	1	M11028	1	M11044	1	M11060	2	M11076	2	M11092	2	M11108	2	M11124	3	M11140	3	M11156	3	M11172	3	M11188	4	M11204	4	M11220	4	M11236	4	M11252
1	M11013	1	M11029	1	M11045	1	M11061	2	M11077	2	M11093	2	M11109	2	M11125	3	M11141	3	M11157	3	M11173	3	M11189	4	M11205	4	M11221	4	M11237	4	M11253
1	M11014	1	M11030	1	M11046	1	M11062	2	M11078	2	M11094	2	M11110	2	M11126	3	M11142	3	M11158	3	M11174	3	M11190	4	M11206	4	M11222	4	M11238	4	M11254
1	M11015	1	M11031	1	M11047	1	M11063	2	M11079	2	M11095	2	M11111	2	M11127	3	M11143	3	M11159	3	M11175	3	M11191	4	M11207	4	M11223	4	M11239	4	M11255

*UNLESS OTHERWISE SPECIFIED TOLERANCES PER DECIMAL PRECISION ARE: X=±1 (±0.039), X.X=±0.5 (±0.020), X.XX=±0.25 (±0.010), X.XXX=±0.127 (±0.005). LEAD SIZE=±0.05 (±0.002), LEAD LENGTH=±0.75 (±0.030). MIN= +DECIMAL PRECISION -0.00 MAX.= +0.00 -DECIMAL PRECISION

LUMEX® Creating LED and LCD Solutions Together™
N. GARY AVE.
CAROL STREAM, IL 60188
PHONE : 800-278-5666
FAX : 630-315-2150
WEB : WWW.LUMEX.COM425

96 * 8 PIXEL, PCB WITH 768 PCS RGB LEDS * 1

DATE :	2017.11.21	DRAWN BY :	C.C.
PAGE :	18 OF 18	CHKD BY :	E.C.
SCALE :	NTF	APRVD BY :	G.Y.
UNIT :	mm [INCH]		Pb