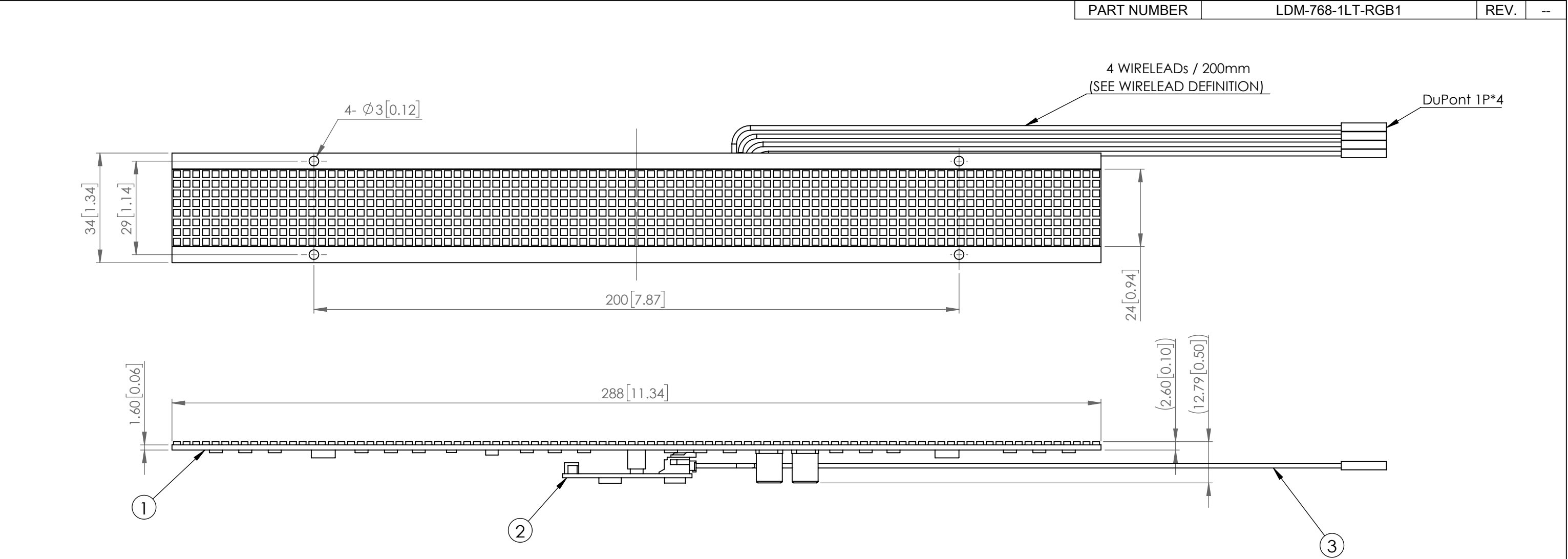




MODULE SPECIFICATION

PARAMETER	SPEC.	UNIT	TEST COND.
PIXEL PITCH	3	mm	--
PIXEL DENSITY	111111	dot/m ²	--
PIXEL FORM	1R1G1B	--	--
VIEWING ANGLE	X=140 / Y=120	degree	--
DRIVE METHOD	1/8	duty	--
MODULE RESOLUTION	96x8	dot	--
LOAD CURRENT	1040 MAX	mA	5V
POWER CONSUMPTION	5.2 MAX	W	5V
BRIGHTNESS	1000 MIN	cd/m2	--
STORAGE TEMPERATURE	-40 TO +60	°C	--
OPERATING TEMPERATURE	-20 TO +60	°C	--

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

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= ^{+DECIMAL PRECISION}_{-0.00} MAX.= ^{+0.00}_{-DECIMAL PRECISION}

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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() fuction, 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	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); }	
*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}							
 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.
		THE SPECIFICATIONS MAY CHANGE AT ANY TIME WITHOUT NOTICE DUE TO NEW MATERIALS OR PRODUCT IMPROVEMENT.				PAGE : 2 OF 14	CHKD BY : E.C.
		CONFIDENTIAL INFORMATION THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF LUMEX INC. EXCEPT AS SPECIFICALLY AUTHORIZED IN WRITING BY LUMEX INC., THE HOLDER OF THIS DOCUMENT SHALL KEEP ALL INFORMATION CONTAINED HEREIN CONFIDENTIAL AND SHALL PROTECT SAME IN WHOLE OR IN PART FROM DISCLOSURE AND DISSEMINATION TO ALL THIRD PARTIES.				SCALE : NTF	APRVD BY : G.Y.
						UNIT : mm [INCH]	

						PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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)				
0x84	Dsisplay 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); }				
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); }				
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); }				

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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)		
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); }		
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); }		

<

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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)		
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); }		

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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); }		

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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); }		

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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); }		

</

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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> AT aa =(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> AT ab =(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> AT d0 =()	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> AT d1 =()	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> AT d2 =(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> AT d3 =(20)	Write_AT_Command("AT d3 =(20)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }		

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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); }		

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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); }		
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); }		

<

				PART NUMBER	LDM-768-1LT-RGB1	REV.	--
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)		
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. ATef=(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("ATef=(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. ATf0=() 2. Wait until receive a module available byte ('E') from ezDisplay 3. Wait 2ms <example> ATf0=()	Write_AT_Command("ATf0=()") 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. ATf1=() 2. Wait until receive a module available byte ('E') from ezDisplay 3. Wait 2ms <example> ATf1=()	Write_AT_Command("ATf1=()") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }		
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. ATf2=(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("ATf2=(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. ATf6=(instruction mode) 2. Wait until receive a module available byte ('E') from ezDisplay 3. Wait 2ms <example> ATf6=(0)	Write_AT_Command("ATf6=(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. ATf7=(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("ATf7=(0)") void Write_AT_Command(char *string) { Serial.print(string); while (Serial.read() != 'E') {} delay(2); }		

</

COLOR CODE Table

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111

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 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 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 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	

*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}