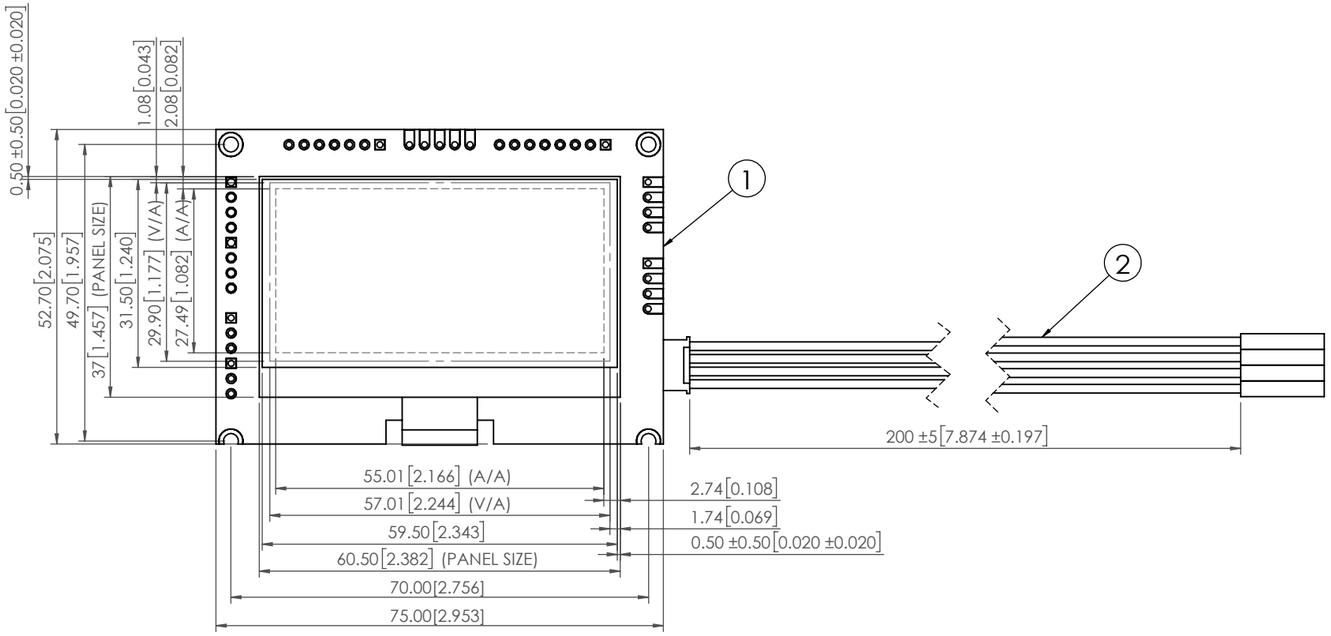
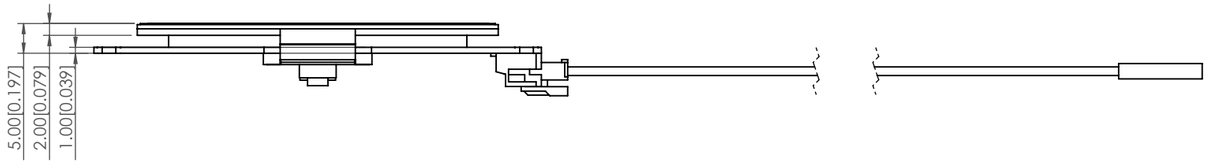
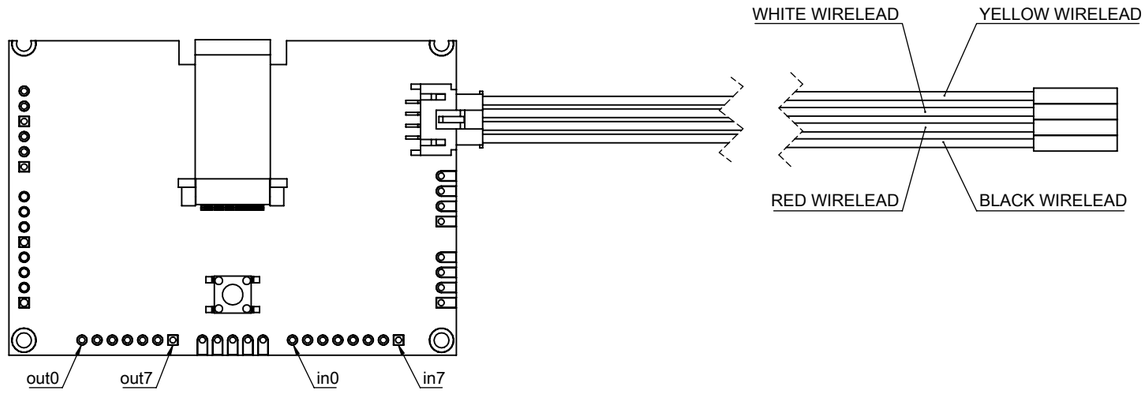
PART NUMBER	LOD-H12864GP-X-UR	REV.	--







BOM :

P/N	ITEM	COMPONENT	QTY
LOD-H12864GP-X-UR	1	LOD-H12864GP-X	1
	2	WIRE002	1

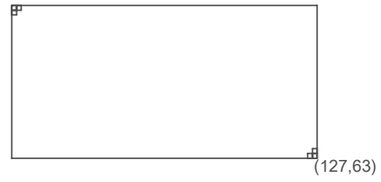
P/N INFORMATION :

PART NUMBER	SIZE	PIXEL COLOR
LOD-H12864GP-W-UR	128x64	WHITE
LOD-H12864GP-Y-UR	128x64	YELLOW
LOD-H12864GP-G-UR	128x64	GREEN
LOD-H12864GP-B-UR	128x64	BLUE

MECHANICAL SPECIFICATIONS :

ITEM	DESCRIPTION
NUMBER OF PIXELS	128 * 64
PANEL SIZE	60.50*37.00*2.00(mm)
ACTIVE AREA	55.01*27.49(mm)
PIXEL PITCH	0.43*0.43(mm)
PIXEL SIZE	0.40*0.40(mm)
WEIGHT	8.60(g)

PIXEL COORDINATE :



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

128x64 PIXELS UART OLED MODULE

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

DATE : 2016/03/08	DRAWN BY : E.C.
PAGE : 1 OF 11	CHKD BY : K.C.
SCALE : NTF	APRVD BY : R.C.
UNIT : mm [INCH]	Pb

ELECTRICAL/OPTO CHARACTERISTICS :

ITEM		SYMBOL	CONDITION	STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
SUPPLY VOLTAGE FOR LOGIC		V_{DD} - V_{SS}	-	4.5	5.0	5.5	V
INPUT VOLTAGE	HIGH	V_{IH}	-	2.64	-	3.3	V
	LOW	V_{IL}	-	0	-	0.66	V
OUTPUT VOLTAGE	HIGH	V_{OH}	-	2.97	-	3.3	V
	LOW	V_{OL}	-	0	-	0.33	V
OLED DRIVING CURRENT		I_{DD}	-	-	50	-	mA
VIEW ANGLE		-	-	160	-	-	deg
DARK ROOM CONTRAST		-	-	-	10000:1	-	-
BRIGHINESS W/POLARIZER		-	-	60	80	-	cd/m^2

*BRIGHINESS=80cd/m^2 , Ta=25℃ , 60% RH , ALL PIXEL ON

UART CONFIGURATION :

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

COMMAND LIST :

Code	Function	Sequence of HEX command mode through UART	API for Arduino

N/A	Sent a page(128X64 bitmap) to OLED
	(An array consist of 1024 bytes bitmap information)
		1. A "for" loop to send 1024 bytes user define display information
		2. Wait until receive a module available byte ('E') from OLED
			for (i = 0 ; i < 1024; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}

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 OLED
			void Write_5X7_Character(int line, int column, int negative, char Char) {
			Serial.write(0x80);
			Serial.write(line);
			Serial.write(column);
			Serial.print(Char);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Write_5X7_String(int line, int column, int negative, char * string) {
			Serial.write(0x81);
			Serial.write(line);
			Serial.write(column);
			Serial.print(string);
			while (Serial.read() !='E') {}
			}

0x82	Write a 8X16 Character	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 OLED
			void Write_8X16_Character(int line, int column, int negative, char Char)
			{
			Serial.write(0x82);
			Serial.write(line);
			Serial.write(column);
			Serial.print(Char);
			while (Serial.read() !='E') {}
			}

0x83	Write a 8X16 String	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 OLED
			void Write_8X16_String(int line, int column, int negative, char * string) {
			Serial.write(0x83);
			Serial.write(line);
			Serial.write(column);
			Serial.print(string);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Write_8X8_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0x84);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

0x85	Dsiplay a 8X16 pattern	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 OLED
			void Write_8X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0x85);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

*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

Code	Function	Sequence of HEX command mode through UART	API for Arduino

0x86	Dsiplay a 16X16 pattern	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 OLED
			void Write_16X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0x86);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

0x87	Dsiplay a 32X32 pattern	1. Send 0x87
		2. Send the Up Left X coordinate of pattern
		3. Send the Up Left Y coordinate of pattern
		5. Send the ID of pattern
		5.Wait until receive a module available byte ('E') from OLED
			void Write_32X32_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0x87);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		7.Wait until receive a module available byte ('E') from OLED
			void Draw_Line(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) {
			Serial.write(0x90);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		7.Wait until receive a module available byte ('E') from OLED
			void Draw_Rectangle(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) {
			Serial.write(0x91);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		7.Wait until receive a module available byte ('E') from OLED
			void Draw_Filled_Rectangle(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) {
			Serial.write(0x92);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(X1_Pos);
			Serial.write(Y1_Pos);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Square(int X0_Pos, int Y0_Pos, int width, int negative) {
			Serial.write(0x93);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Circle(int X0_Pos, int Y0_Pos, int radius, int negative) {
			Serial.write(0x94);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(radius);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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	Sequence of HEX command mode through UART	API for Arduino

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Filled_Circle(int X0_Pos, int Y0_Pos, int radius, int negative) {
			Serial.write(0x95);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(radius);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Triangle_Up_Ward(int X0_Pos, int Y0_Pos, int height, int negative) {
			Serial.write(0x96);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Filled_Triangle_Up_Ward(int X0_Pos, int Y0_Pos, int height, int negative) {
			Serial.write(0x97);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Triangle_Down_Ward(int X0_Pos, int Y0_Pos, int height, int negative) {
			Serial.write(0x98);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw__Filled_Triangle_Down_Ward(int X0_Pos, int Y0_Pos, int height, int negative)
			{
			Serial.write(0x99);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(height);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int negative) {
			Serial.write(0x9a);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Filled_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int negative) {
			Serial.write(0x9b);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

Code	Function	Sequence of HEX command mode through UART	API for Arduino

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int negative) {
			Serial.write(0x9c);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 1 or 0 for display mode (1 for positive, 0 for negative)
		6.Wait until receive a module available byte ('E') from OLED
			void Draw_Filled_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int negative) {
			Serial.write(0x9d);
			Serial.write(X0_Pos);
			Serial.write(Y0_Pos);
			Serial.write(width);
			Serial.write(0 or 1);
			while (Serial.read() !='E') {}
			}

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 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') {}
			}

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 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') {}
			}

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 OLED
			void Display_Row_By_Row_Up_Ward(int Speed) {
			Serial.write(0xa0);
			Serial.write(speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Display_Row_By_Row_Down_Ward(int speed) {
			Serial.write(0xa1);
			Serial.write(speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Display_Column_By_Column_Left_Ward(int speed) {
			Serial.write(0xa2);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

*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.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

Code	Function	Sequence of HEX command mode through UART	API for Arduino

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 OLED
			void Display_Column_By_Column_Right_Ward(int Speed) {
			Serial.write(0xa2);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Erase_Row_By_Row_Up_Ward(int Speed) {
			Serial.write(0xa4);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Erase_Row_By_Row_Down_Ward(int Speed) {
			Serial.write(0xa5);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Erase_Column_By_Column_Left_Ward(int Speed) {
			Serial.write(0xa6);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Erase_Column_By_Column_Right_Ward(int Speed) {
			Serial.write(0xa7);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Display_Inside_Out(int Speed) {
			Serial.write(0xa8);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Display_Outside_In(int Speed) {
			Serial.write(0xa9);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

*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.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

Code	Function	Sequence of HEX command mode through UART	API for Arduino

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 OLED
			void Erase_Inside_Out(int Speed) {
			Serial.write(0xaa);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

0xab	Erase image Outside In	1. Send 0xab
		2. Send the speed (typical time is 20ms)
		6.Wait until receive a module available byte ('E') from OLED
			void Erase_Outside_In(int Speed) {
			Serial.write(0xab);
			Serial.write(Speed);
			while (Serial.read() !='E') {}
			}

0xc0	Build user define 8X8 pattern bitmap into OLED's display memory
	(Maximun number of user define 8X8 pattern is 10 (0~9))
		1. Send 0xc0
		2. Send the pattern ID
		3. Sent the bitmap of this pattern ID
		4.Wait until receive a module available byte ('E') from OLED
			void Build_User_Define_8X8_Pattern() {
			Serial.write(0xc0);
			Serial.write(0);
			for (i = 0; i < 8; i++)
			{
			Serial.write(User_Define_8X8_pattern_ID[i]);
			}
			while (Serial.read() !='E') {}
			}

0xc1	Build user define 8X16 pattern bitmap into OLED's display memory
	(Maximun number of user define 8X16 pattern is 10 (0~9))
		1. Send 0xc1
		2. Send the pattern ID
		3. Sent the bitmap of this pattern ID
		4.Wait until receive a module available byte ('E') from OLED
			void Build_User_Define_8X16_Pattern() {
			Serial.write(0xc1);
			Serial.write(0);
			for (i = 0; i < 16; i++)
			{
			Serial.write(User_Define_8X16_pattern_ID[i]);
			}
			while (Serial.read() !='E') {};
			}

0xc2	Build user define 16X16 pattern bitmap into OLED's display memory
	(Maximun number of user define 16X16 pattern is 10 (0~9))
		1. Send 0xc2
		2. Send the pattern ID
		3. Sent the bitmap of this pattern ID
		4.Wait until receive a module available byte ('E') from OLED
			void Build_User_Define_16X16_Pattern() {
			Serial.write(0xc2);
			Serial.write(0);
			for (i = 0; i < 32; i++)
			{
			Serial.write(User_Define_16X16_pattern_ID[i]);
			}
			while (Serial.read() !='E') {}
			}

0xc3	Build user define 32X32 pattern bitmap into OLED's display memory
	(Maximun number of user define 32X32 pattern is 5 (0~4))
		1. Send 0xc3
		2. Send the pattern ID
		3. Sent the bitmap of this pattern ID
		4.Wait until receive a module available byte ('E') from OLED
			void Build_User_Define_32X32_Pattern() {
			Serial.write(0xc3);
			Serial.write(0);
			for (i = 0; i < 128; i++)
			{
			Serial.write(User_Define_32X32_pattern_ID[i]);
			}
			while (Serial.read() !='E') {}
			}

0xc4	Dsiplay a user define 8X8 pattern
	(Build user define 8X8 pattern function needs to run before this function)
		1. Send 0xc4
		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 OLED
			void Write_User_Define_8X8_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0xc4);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

Code	Function	Sequence of HEX command mode through UART	API for Arduino

0xc5	Dsiplay a user define 8X16 pattern
	(Build user define 8X16 pattern function needs to run before this function)
		1. Send 0xc5
		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 OLED
			void Write_User_Define_8X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0xc5);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

0xc6	Dsiplay a user define 16X16 pattern
	(Build user define 16X16 pattern function needs to run before this function)
		1. Send 0xc6
		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 OLED
			void Write_User_Define_16X16_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0xc6);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

0xc7	Dsiplay a user define 32X32 pattern
	(Build user define 32X32 pattern function needs to run before this function)
		1. Send 0xc7
		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 OLED
			void Write_User_Define_32X32_Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) {
			Serial.write(0xc7);
			Serial.write(Up_Left_Xpos);
			Serial.write(Up_Left_Ypos);
			Serial.write(Pattern_ID);
			while (Serial.read() !='E') {}
			}

0xd0	Clear display	1. Send 0xd0
		2.Wait until receive a module available byte ('E') from OLED
			void Clear_Display_Momery(void) {
			Serial.write(0xd0);
			while (Serial.read() !='E') {}
			}

0xd1	Show the data in the display memory	1. Send 0xd1
		2.Wait until receive a module available byte ('E') from OLED
			void Show_Display_Momery(void) {
			Serial.write(0xd1);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Scroll_Whole_Display_Memory_Up(int shift time) {
			Serial.write(0xd2);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Scroll_Whole_Display_Memory_Down(int shift time) {
			Serial.write(0xd3);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			}

Code	Function	Sequence of HEX command mode through UART	API for Arduino

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 OLED
			void Scroll_Whole_Display_Memory_Left(int shift time) {
			Serial.write(0xd4);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			}

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 OLED
			void Scroll_Whole_Display_Memory_Right(int shift time) {
			Serial.write(0xd5);
			Serial.write(shift time);
			while (Serial.read() !='E') {}
			}

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 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') {};
			}

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 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') {}
			}

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 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') {}
			}

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 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') {}
			}

0xf0	Turn display Off	1. Send 0xf0
		2.Wait until receive a module available byte ('E') from OLED
			void Display_Off(void){
			Serial.write(0xf0);
			while (Serial.read() !='E') {}
			}

*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.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

Code	Function	Sequence of HEX command mode through UART	API for Arduino

0xf1	Turn display On	1. Send 0xf1
		2.Wait until receive a module available byte ('E') from OLED
			void Display_On(void) {
			Serial.write(0xf1);
			while (Serial.read() !='E') {}
			}

0xf2	Set the brightness of the OLED	1. Send 0xf2
		2. Send the level of brightness
		3.Wait until receive a module available byte ('E') from OLED
			void Set_Display_Contrast(int contrast) {
			Serial.write(0xf2);
			Serial.write(contrast);
			while (Serial.read() !='E') {}
			}

0xf3	Set the status of 8 output pins on OLED	1. Send 0xf3
		2. Send the output pin No.
		3. Send 0 or 1 (0--> Low, 1-->High)
		4.Wait until receive a module available byte ('E') from OLED
			void Set_Output_Port(int Output_pin_No, int HL) {
			Serial.write(0xf3);
			Serial.write(Output_pin_No);
			Serial.write(HL);
			while (Serial.read() !='E') {}
			}

0xf4	Read the input pins status on the OLED	1. Send 0xf4
		2. Send the input pin No.
		4. Recive the input pins status from OLED (0 or 1)
		5. Return the input pins status
			int Read_Input_Port(Input_pin_No) {
			Serial.write(0xf4);
			Serial.write(Input_pin_No);
			while (Serial.available() <= 0) {}
			incomingByte = Serial.read();
			return incomingByte;
			}

0xf6	Change Instruction mode
	(1 for AT command)
		1. Send 0xf6
		2. Send instruction mode 1
		3. Wait until receive a module available byte ('E') from OLED
			int Change_Display_Mode(int mode) {
			Serial.write(0xf6);
			Serial.write(1);
			while (Serial.read() !='E') {}
			}

*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.= $^{+0.00}_{-\text{DECIMAL PRECISION}}$

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	°C
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

128x64 PIXELS UART OLED MODULE

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

DATE : 2016/03/08	DRAWN BY : E.C.
PAGE : 11 OF 11	CHKD BY : K.C.
SCALE : NTF	APRVD BY : R.C.
UNIT : mm [INCH]	Pb