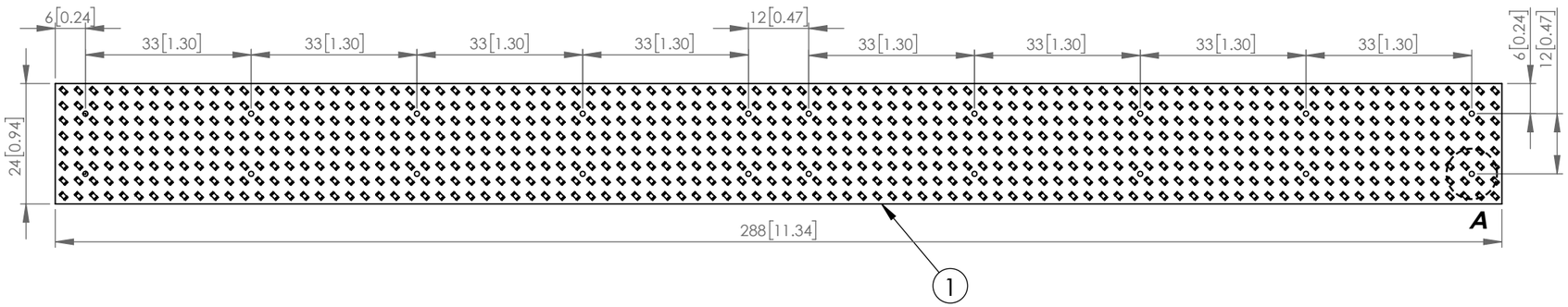
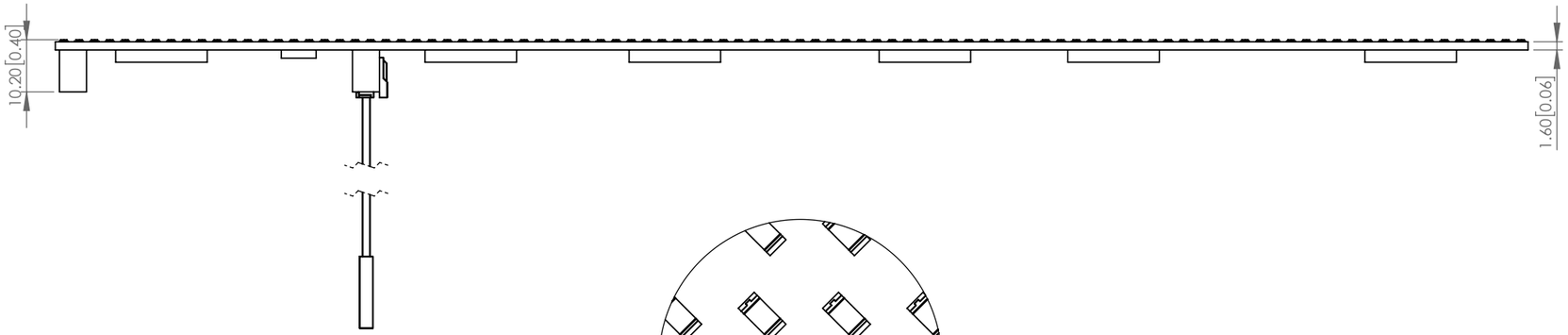
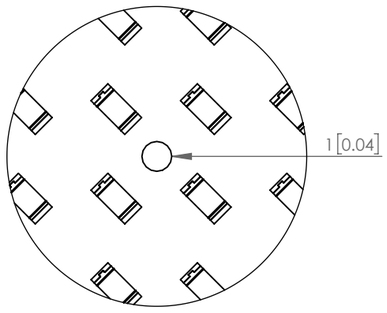
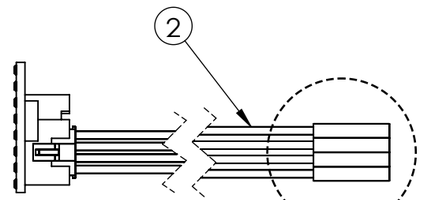
PART NUMBER	LDM-768-1LT-X1	REV.	--

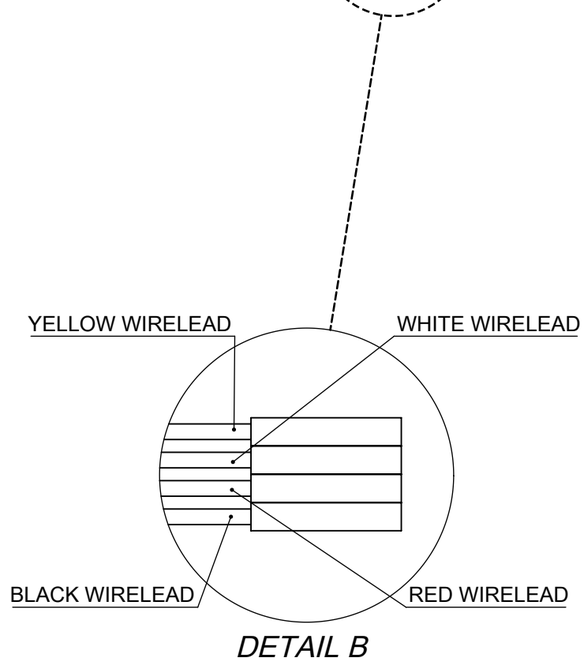






DETAIL A





DETAIL B

*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 PIXELS, PCB WITH 768 PCS LEDS * 1

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

DATE : 2016/09/28	DRAWN BY : E.C.
PAGE : 1 OF 8	CHKD BY : K.C.
SCALE : NTF	APRVD BY : R.C.
UNIT : mm [INCH]	Pb

BOM :

P/N	ITEM	COMPONENT	QTY
LDM-768-1LT-X	1	LDM-768-1LT-X1-PCB	1
	2	WIRE002	1

P/N INFORMATION :

PART NUMBER	COLOR
LDM-768-1LT-G1	GREEN
LDM-768-1LT-Y1	YELLOW
LDM-768-1LT-R1	RED

WIRELEAD DEFINITION :

COLOR	DEFINITION
YELLOW	TX1
WHITE	RX1
RED	5V
BLACK	GND

LOAD CURRENT & POWER CONSUMPTION WITH ALL LED ON :

Current consumptiom	GREEN	YELLOW	RED	UNIT	GREEN	YELLOW	RED	UNIT
All LEDs off	28	28	28	mA	0.14	0.14	0.14	W
Diming level 0	128	302	308	mA	0.64	1.51	1.54	W
Diming level 1	228	420	440	mA	1.14	2.1	2.2	W
Diming level 2	320	550	590	mA	1.6	2.75	2.95	W
Diming level 3	420	670	710	mA	2.1	3.35	3.55	W
Diming level 4	510	790	830	mA	2.55	3.95	4.15	W
Diming level 5	610	900	950	mA	3.05	4.5	4.75	W
Diming level 6	690	1000	1060	mA	3.45	5	5.3	W
Diming level 7	790	1110	1180	mA	3.95	5.55	5.9	W
Diming level 8	870	1200	1290	mA	4.35	6	6.45	W
Diming level 9	950	1310	1390	mA	4.75	6.55	6.95	W
Diming level 10	1010	1390	1490	mA	5.05	6.95	7.45	W
Diming level 11	1115	1490	1590	mA	5.575	7.45	7.95	W

UART CONFIGURATION :

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

LED ELECTRO-OPTICAL CHARACTERISTICS TA =25° :

PARAMETER		MIN	TYP	MAX	UNITS	TEST COND
GREEN LED	PEAK WAVELENGTH		525		nm	If=20mA
	FORWARD VOLTAGE	2.7	3.3	3.7	Vf	If=20mA
	REVERSE VOLTAGE			5.0	Vr	Ir=20uA
	LUMINOUS INTENSITY	140		450	mcd	If=20mA
	VIEWING ANGLE		120		2x theta1/2	If=20mA
	EMITTED COLOR	GREEN				
	EPOXY LENS FINISH	WATER CLEAR				
YELLOW LED	PEAK WAVELENGTH		591		nm	If=20mA
	FORWARD VOLTAGE	1.7	2.0	2.4	Vf	If=20mA
	REVERSE VOLTAGE			5.0	Vr	Ir=20uA
	LUMINOUS INTENSITY	16	40		mcd	If=20mA
	VIEWING ANGLE		100		2x theta1/2	If=20mA
	EMITTED COLOR	YELLOW				
	EPOXY LENS FINISH	WATER CLEAR				
RED LED	PEAK WAVELENGTH		632		nm	If=20mA
	FORWARD VOLTAGE	1.7	2.0	2.4	Vf	If=20mA
	REVERSE VOLTAGE			5.0	Vr	Ir=20uA
	LUMINOUS INTENSITY	37	56		mcd	If=20mA
	VIEWING ANGLE		100		2x theta1/2	If=20mA
	EMITTED COLOR	RED				
	EPOXY LENS FINISH	WATER CLEAR				

LED LIMITS OF SAFE OPERATION AT 25° :

PARAMETER		MAX	UNITS
GREEN LED	PEAK FORWARD CURRENT	100	mA
	FORWARD CURRENT	25	mA
	POWER DISSIPATION	95	mW
	ELECTROSTATIC DISCHARGE	150	V
	OPERATING TEMP	-40~+85	°C
	STORAGE TEMP	-40~+90	°C
	SOLDERING TEMP	MAX +260°C @3 SEC	
YELLOW LED	PEAK FORWARD CURRENT	60	mA
	FORWARD CURRENT	25	mA
	POWER DISSIPATION	60	mW
	ELECTROSTATIC DISCHARGE	2000	V
	OPERATING TEMP	-40~+85	°C
	STORAGE TEMP	-40~+90	°C
	SOLDERING TEMP	MAX +260°C @3 SEC	
RED LED	PEAK FORWARD CURRENT	60	mA
	FORWARD CURRENT	25	mA
	ELECTROSTATIC DISCHARGE	2000	V
	POWER DISSIPATION	60	mW
	OPERATING TEMP	-40~+85	°C
	STORAGE TEMP	-40~+90	°C
	SOLDERING TEMP	MAX +260°C @3 SEC	

*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 PIXELS, PCB WITH 768 PCS LEDS * 1

```
void Write_AT_Command(char *string)
{
  Serial.print(string);
  while (Serial.read() != 'E') {}
  delay(2);
}
```

COMMAND LIST :

Code	Function	Instruction of AT Command mode	API for Arduino	API of using Write_AT_Command() subroutine above

N/A	Sent a image(192X64 bitmap) to LED Display (An array consist of 1536 bytes bitmap information)	1. A ""for"" loop to send 1536 bytes user define display information
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			for (i = 0 ; i < 1536; i++)
			{
			Serial.write(User_define_array[i]);
			}
			while (Serial.read() !='E') {}
			delay(2);	

0x80	Write a 5X7 Character	1. AT80=(line,column,Character)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT80=(0,0,A)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT80=(0,0,A)")

0x81	Write a 8X8 String	1.AT81=(line,column,String)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT81=(0,0,ABCD1234)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT81=(0,0,ABCD1234)")

0x82	Write a 8X16 Character	1.AT82=(line,column,Character)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT82=(0,0,A)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT82=(0,0,A)")

0x83	Write a 8X16 String	1.AT83=(line,column,String)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT83=(0,0,ABCD1234)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT83=(0,0,ABCD1234)")

0x84	Dsiplay a 8X8 pattern	1. AT84=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT84=(16,32,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT84=(16,32,1)")

0x85	Dsiplay a 8X16 pattern	1.AT85=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT85=(16,32,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT85=(16,32,1)")

0x86	Dsiplay a 16X16 pattern	1. AT86=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT86=(16,32,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT86=(16,32,1)")

0x87	Dsiplay a 32X32 pattern	1. AT87=(X position,Y position,pattern ID)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT87=(16,32,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT87=(16,32,1)")

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	Instruction of AT Command mode	API for Arduino	API of using Write_AT_Command() subroutine above

0x90	Draw a line	1. AT90=(X0 position,Y0 position,X1 position,Y1 position,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT90=(0,0,127,63,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT90=(0,0,127,63,1)")

0x91	Draw a Rectangle	1. AT91=(X0 position,Y0 position,X1 position,Y1 position,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT91=(10,10,100,49,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT91=(10,10,100,49,1)")

0x92	Draw a filled Rectangle	1. AT92=(X0 position,Y0 position,X1 position,Y1 position,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT92=(10,10,100,49,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT92=(10,10,100,49,1)")

0x93	Draw a Square	1. AT93=(X position,Y position,Width,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT93=(8,10,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT93=(8,10,30,1)")

0x94	Draw a Circle	1. AT94=(X position,Y position,Radius,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT94(64,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT94(64,32,30,1)")

0x95	Draw a filled Circle	1. AT95=(X position,Y position,Radius,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT95=(64,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT95=(64,32,30,1)")

0x96	Draw a tip upward Triangle	1. AT96=(X position,Y position,Height,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT96=(64,10,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT96=(64,10,30,1)")

0x97	Draw a filled tip upward Triangle	1. AT97=(X position,Y position,Height,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT97=(64,10,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT97=(64,10,30,1)")

0x98	Draw a tip downward Triangle	1. AT98=(X position,Y position,Height,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT98=(64,50,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT98=(64,50,30,1)")

0x99	Draw a filled tip downward Triangle	1. AT99=(X position,Y position,Height,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT99=(64,50,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT99=(64,50,30,1)")

0x9a	Draw a tip leftward Triangle	1. AT9a=(X position,Y position,Width,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9a=(16,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9a=(16,32,30,1)")

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	Instruction of AT Command mode	API for Arduino	API of using Write_AT_Command() subroutine above

0x9b	Draw a filled tip leftward Triangle	1. AT9b=(X position,Y position,Width,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9b=(16,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9b=(16,32,30,1)")

0x9c	Draw a tip rightward Triangle	1. AT9c=(X position,Y position,Width,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9c=(120,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9c=(120,32,30,1)")

0x9d	Draw a filled tip rightward Triangle	1. AT9d=(X position,Y position,Width,0 or 1)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9d=(120,32,30,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9d=(120,32,30,1)")

0x9e	Set a pixel for positive display (show pixel)	1. AT9e=(X position,Y position)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9e=(120,32)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9e=(120,32)")

0x9f	Set a pixel for negative display (clear pixel)	1. AT9f=(X position,Y position)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("AT9f=(120,32)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("AT9f=(120,32)")

0xa0	Display image row by row Up Ward	1. ATa0=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa0=(20)")
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa0=(20)")

0xa1	Display image row by row Down Ward	1. ATa1=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa1=(20)")
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa1=(20)")

0xa2	Display image column by column Left Ward	1. ATa2=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa2=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa2=(20)")

0xa3	Display image column by column Right Ward	1. ATa3=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa3=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa3=(20)")

0xa4	Erase image row by row Up Ward	1. ATa4=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa4=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa4=(20)")

0xa5	Erase image row by row Down Ward	1. ATa5=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa5=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa5=(20)")

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	Instruction of AT Command mode	API for Arduino	API of using Write_AT_Command() subroutine above

0xa6	Erase image column by column Left Ward	1. ATa6=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa6=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa6=(20)")

0xa7	Erase image column by column Right Ward	1. ATa7=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa7=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa7=(20)")

0xa8	Display image Inside Out	1. ATa8=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa8=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa8=(20)")

0xa9	Display image Outside In	1. ATa9=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATa9=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATa9=(20)")

0xaa	Erase image Inside Out	1. ATaa=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATaa=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATaa=(20)")

0xab	Erase image Outside In	1. ATab=(Speed in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATab=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATab=(20)")

0xd0	Clear display	1. ATd0=()
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd0=()");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd0=()")

0xd1	Show the data in the display memory	1. ATd1=()
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd1=()");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd1=()")

0xd2	Scroll the whole display upward	1. ATd2=(shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd2=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd2=(20)")

0xd3	Scroll the whole display downward	1. ATd3=(shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd3=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd3=(20)")

0xd4	Scroll the whole display leftward	1. ATd4=(shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd4=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd4=(20)")

Code	Function	Instruction of AT Command mode	API for Arduino	API of using Write_AT_Command() subroutine above

0xd5	Scroll the whole display rightward	1. ATd5=(shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd5=(20)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd5=(20)")

0xd6	Scroll the section display upward	1. ATd6=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd6=(10,16,120,50,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd6=(10,16,120,50,1)")

0xd7	Scroll the section display downward	1. ATd7=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd7=(10,16,120,50,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd7=(10,16,120,50,1)")

0xd8	Scroll the section display leftward	1. ATd8=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd8=(10,16,120,50,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd8=(10,16,120,50,1)")

0xd9	Scroll the section display rightward	1. ATd9=(X0 position,Y0 position,X1 position,Y1 position, shif time in ms)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATd9=(10,16,120,50,1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATd9=(10,16,120,50,1)")

0xda	Display quarter of display memory
	(Available for Mode0, 1, and 2 only)
		1. ATda=(Quadrant 0~3)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATda=(1)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATda=(1)")

0xf0	Turn display Off	1. ATf0=()
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf0=()");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf0=()")

0xf1	Turn display On	1. ATf1=()
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf1=()");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf1=()")

0xf2	Set the brightness of the LED Module	1. ATf2=(levele of brightness 0~11)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf2=(5)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf2=(5)")

0xf3	Inverse image	1. ATf3=()
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf3=()");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf3=()")

0xf6	Change Instruction mode
	(0 for Hex Coammand, 1 for AT Command)
		1. ATf6=(0)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf6=(0)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf6=(0)")

0xf7	Change Display Mode	1. ATf7=(Display Mode)
		2. Wait until receive a module available byte ('E') from LED Display
		3. Wait 2ms
			Serial.print("ATf7=(0)");
			while (Serial.read() !='E') {}
			delay(2);
				Write_AT_Command("ATf7=(0)")

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	℉
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 PIXELS, PCB WITH 768 PCS LEDS * 1

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

DATE : 2016/09/28	DRAWN BY : E.C.
PAGE : 8 OF 8	CHKD BY : K.C.
SCALE : NTF	APRVD BY : R.C.
UNIT : mm [INCH]	Pb

Общество с ограниченной ответственностью «МосЧип» ИНН 7719860671 / КПП 771901001
Адрес: 105318, г.Москва, ул.Щербаковская д.3, офис 1107

Данный компонент на территории Российской Федерации

Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

http://moschip.ru/get-element

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:
moschip.ru
moschip.ru_4
moschip.ru_6
moschip.ru_9